A UNIFIED GAME-THEORETIC METHODOLOGY FOR THE JOINT LOAD SHARING, ROUTING AND CONGESTION CONTROL PROBLEM

Volume II

by

Anastasios A. Economides

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Computer Engineering)

December 1990

Chapter 5

Dynamic Formulation

In this chapter, we develop three novel methodologies for the dynamic problem: i) the dynamic team optimization methodology, ii) the dynamic Nash game methodology, and iii) the dynamic Stackelberg game methodology. For each methodolodology, we develop three alternative formulations of the joint problem, namely an optimal control, a nonlinear complementarity problem and a variational inequality formulation. For each formulation, we state the necessary and sufficient conditions for existence and uniqueness of the solution. From Pontryagin's maximum principle, we also derive the form of the solution, that there should be flow only on minimum length paths, to minimum length destinations, The length at each system resource is appropriately defined for each case. Then we apply these three methodologies to datagram, virtual circuit and integrated services networks. We develop new dynamic queueing models for multiple classes and priority classes of jobs, as well as linearized approximate dynamic queueing models and Wiener process models. We introduce several new cost functions and state constraints. We explicitly solve an example for virtual circuit networks. We consider a virtual circuit network with Poisson arrivals of virtual circuits and packets, and exponential service requirements. We want to minimize the expected cost of servicing or rejecting virtual circuits, minimize the expected cost of packet delay and maximize the expected profit from packet throughput. We find the dynamic team optimality conditions and we propose a state dependent routing and congestion control algorithm. We investigate and compare (via simulation) this state dependent routing algorithm to the optimal quasi-static algorithm. We find that the more often that we update the state dependent algorithm and the more recent information that we use the better. When the updating period is not much larger than the mean interarrival time of virtual circuits, then this state dependent algorithm achieves smaller average packet delay than the optimal quasi-static algorithm.

5.1 Team Optimal Solution

In this section, we formulate the dynamic joint load sharing, routing and congestion control problem on the path flow space as a cooperative dynamic team game among cooperative classes.

Customers of each class cooperate in using the resources of the distributed system for the social welfare. The behavior of each class is similar to that of any other class, that is to operate optimally for the average job. Ho [218] presents a tutorial on team theory where the decision makers have access to different information concerning the underline uncertainties. Leitmann [297] provides a rigorous analysis of cooperative and zero-sum non-cooperative games.

Next, we give the definition for a Pareto optimal solution, for the joint load sharing, routing and congestion control problem on the path flows.

Definition:

A vector $(\mathbf{\Phi}^*, \mathbf{\Psi}^*) \in (\mathbf{RC}, \mathbf{LS})$ *is called a* <u>*Pareto optimal solution*</u> *for a* C*-class joint load sharing, routing and congestion control problem if and only if there exists no other vector* $(\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$ *such that*

$J^c(\mathbf{\Phi}, \mathbf{\Psi}) \quad \leq J^c(\mathbf{\Phi}^*, \mathbf{\Psi}^*) \quad \forall\ (\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$

with strict inequality holding for at least one class c.

Define a global cost function

$$J(\mathbf{\Phi}, \mathbf{\Psi}) = \left[\sum_c [w^c * J^c(\mathbf{\Phi}, \mathbf{\Psi})]^p \right]^{1/p}$$

where $1 \leq p < \infty, \quad \sum_{c=1}^{C} w^c = 1, \quad w^c \geq 0 \ \ \forall\ c.$

For $p \to \infty$, we have a minimax problem [122], since the cost function becomes

$$J(\mathbf{\Phi}, \mathbf{\Psi}) = \max_{c} \{w^c * J^c(\mathbf{\Phi}, \mathbf{\Psi})\}$$

Another problem formulation is

$$\min_{\epsilon, \mathbf{\Phi}, \mathbf{\Psi}} \ \epsilon$$

such that

$$w^c * J^c(\mathbf{\Phi}, \mathbf{\Psi}) \leq \epsilon \quad \forall\ c$$

.

Furthermore, another problem formulation is

$$\min_{\mathbf{\Phi}, \mathbf{\Psi}} \ J(\mathbf{\Phi}, \mathbf{\Psi})$$

such that

$$J^c(\mathbf{\Phi}, \mathbf{\Psi}) \leq \hat{J}^c(\mathbf{\Phi}, \mathbf{\Psi}) \quad \forall\ c$$

.

where $\hat{J}^c$ is the maximum acceptable value for the cost function J^c.

Next, we give the definition for a team optimal solution [27], for the joint load sharing, routing and congestion control problem on the path flows.

Definition:

A vector $(\mathbf{\Phi}^*, \mathbf{\Psi}^*) \in (\mathbf{RC}, \mathbf{LS})$ is called a team optimal solution for a C-class joint load sharing, routing and congestion control problem if and only if

$$\mathrm{J}\ (\Phi^*, \mathbf{\Psi}^*) \quad \leq J(\mathbf{\Phi}, \mathbf{\Psi}) \quad \forall\ (\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$$

In the next sections, we develop three alternative formulations for the joint load sharing, routing and congestion control problem.

5.1.1 Optimal Control Formulation

In this section, we formulate the dynamic cooperative joint load sharing, routing and congestion control problem as an Optimal Control Problem (OCP). Algorithms for solving OCPs is a thoroughly investigated research area and popular algorithms may be found in books by Athans & Falb [14], Lee & Markus [292], Plant [381], Sage [415], McCausland [325], Dyer & McReynolds [131], Kirk [254], Russell [412], Gruver & Sachs [203], Sethi & Thompson [440], Knowles [262], Lewis [301] among others.

Define the Hamiltonian as

$$H(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}) = g(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) + \mathbf{P} * \mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$$

where $\mathbf{P} = [\dots\ P^{c,k}_{ij[sd]}\ \dots\ P^{c,k}_{i[sd]}\ \dots\ P^{c,k}_{o[sd]}\ \dots\ P^{c,k}_{[.d][sd]}\ \dots]$: vector of costate variables.

Define also the derivatives of H with respect to the congestion, routing and load sharing fractions at $(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))$ as

$$\frac{\partial H^*}{\phi^c_{o[sd]}} = \frac{\partial H(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P})}{\phi^c_{o[sd]}} \Big|_{(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))}$$

$$\frac{\partial H^*}{\phi^c_{\pi[sd]}} = \frac{\partial H(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P})}{\phi^c_{\pi[sd]}} \Big|_{(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))}$$

$$\frac{\partial H^*}{\psi^c_{[sd]}} = \frac{\partial H(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P})}{\psi^c_{[sd]}} \Big|_{(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))}$$

Define also the Lagrangian as

$$\begin{aligned} \mathrm{L}\ (\mathrm{t}, \mathbf{X}, \Phi, \mathbf{\Psi}, \mathbf{P}, \mathbf{Q}) &= \mathrm{H}\ (\mathrm{t}, \mathbf{X}, \Phi, \mathbf{\Psi}, \mathbf{P}) + \\ &+ \sum_{c} \sum_{[sd] \in \mathbf{SD}^c} Q^c_{[sd]} * \left[1 - \phi^c_{o[sd]} - \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]} \right] + \\ &+ \sum_{c} \sum_{[s.] \in \mathbf{S}^c} Q^c_{[s.]} * \left[1 - \sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]} \right] \end{aligned}$$

with $\qquad \phi^c_{o[sd]},\ \phi^c_{\pi[sd]},\ \psi^c_{[sd]} \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$

where $\mathbf{Q} = [\dots\ Q^c_{[sd]}\ \dots\ Q^c_{[s.]}\ \dots]$: vector of multipliers for the constraints of the congestion control, routing and load sharing fractions.

Define also the derivatives of L with respect to the congestion, routing and load sharing fractions at $(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))$ as

$$\frac{\partial L^*}{\phi^c_{o[sd]}} = \frac{\partial L(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}, \mathbf{Q})}{\phi^c_{o[sd]}} |_{(t,\mathbf{X}^*(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t),\mathbf{P}(t),\mathbf{Q}(t))}$$

$$\frac{\partial L^*}{\phi^c_{\pi[sd]}} = \frac{\partial L(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}, \mathbf{Q})}{\phi^c_{\pi[sd]}} |_{(t,\mathbf{X}^*(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t),\mathbf{P}(t),\mathbf{Q}(t))}$$

$$\frac{\partial L^*}{\psi^c_{[sd]}} = \frac{\partial L(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}, \mathbf{Q})}{\psi^c_{[sd]}} |_{(t,\mathbf{X}^*(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t),\mathbf{P}(t),\mathbf{Q}(t))}$$

Theorem :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 and final time t_f.

$(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a team optimal solution if and only if it solves the following* *Optimal Control Problem:*

minimize $\displaystyle\int_{t_0}^{t_f} g(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))dt$

with respect to $(\Phi(t), \mathbf{\Psi}(t))$

such that $\dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$

$$\mathbf{X}(\mathrm{t}_0) = \mathbf{X}_0$$

$$(\Phi(t), \mathbf{\Psi}(t)) \in (\mathbf{RC}, \mathbf{LS})$$

Proof: It follows from the definition of the team optimal solution.□

Necessary conditions for optimality are provided by Pontryagin's Maximum Principle. Besides the previously referred books on optimal control theory, some other books that contain material on Pontryagin's maximum principle are the following: Hestenes [215], Arrow & Kurz [12], Tabak & Kuo [476], Boltyanskii

[58], Berkovitz [33], Bryson & Ho [79], Fleming & Rishel [162], Boltyanskii [59], Leitmann [298], Macki & Strauss [315], Alekseev, Tikhomirov & Fomin [8].

Theorem : *necessary conditions*

Consider the dynamic joint load sharing, routing and congestion control problem in distributed-systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f.

Let $g(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *be continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi}) \;\; \forall\, t \in [t_0, t_f]$.

If $(\hat{\mathbf{\Phi}}^*(t, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}_0)) = (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a piecewise continuous open-loop team optimal solution and* $\{\mathbf{X}^*(t),\ t \in [t_0, t_f]\}$ *is the corresponding state trajectory, then* $\exists\ \mathbf{P}(t) : [t_0, t_f] \rightarrow \mathbf{R}^n$ *continuous and piecewise continuously differentiable vector function, such that* $\forall\, t \in [t_0, t_f]$:

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\left[\frac{\partial H^*}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{o[sd]}(t) = 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\frac{\partial H^*}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{\pi[sd]}(t) = 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c, c$$

$$\left[\frac{\partial H^*}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)\right] * \psi^{c*}_{[sd]}(t) = 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\frac{\partial H^*}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^*}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^*}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))$$

$$\mathbf{P}(\mathrm{t}_f) = \mathbf{0}$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d]\in\mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

Proof: The Lagrangian is

$$L = H + \sum_c \sum_{[sd]\in\mathbf{SD}^c} Q^c_{[sd]} * \left[1 - \phi^c_{o[sd]} - \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}\right] +$$
$$+ \sum_c \sum_{[s.]\in\mathbf{S}^c} Q^c_{[s.]} * \left[1 - \sum_{[.d]\in\mathbf{D}^c_{[s.]}} \psi^c_{[sd]}\right]$$

with $\phi^c_{o[sd]},\ \phi^c_{\pi[sd]},\ \psi^c_{[sd]} \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$

Pontryagin's maximum principle necessary conditions are:

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\frac{\partial L^*}{\partial \phi^c_{o[sd]}} * \phi^{c*}_{o[sd]}(t) = 0 \;\Rightarrow\; \left[\frac{\partial H^*}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{o[sd]}(t) = 0$$

$$\forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^*}{\partial \phi^c_{\pi[sd]}} * \phi^{c*}_{\pi[sd]}(t) = 0 \;\Rightarrow\; \left[\frac{\partial H^*}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{\pi[sd]}(t) = 0$$

$$\forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c, c$$

$$\frac{\partial L^*}{\partial \psi^c_{[sd]}} * \psi^{c*}_{[sd]}(t) = 0 \;\Rightarrow\; \left[\frac{\partial H^*}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)\right] * \psi^{c*}_{[sd]}(t) = 0$$

$$\forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\frac{\partial L^*}{\partial \phi^c_{o[sd]}} \geq 0 \;\Rightarrow\; \frac{\partial H^*}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^*}{\partial \phi^c_{\pi[sd]}} \geq 0 \;\Rightarrow\; \frac{\partial H^*}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^*}{\partial \psi^c_{[sd]}} \geq 0 \;\Rightarrow\; \frac{\partial H^*}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))$$

$$\mathbf{P}(\mathrm{t}_f) = \mathbf{0}$$

$$\frac{\partial L^*}{\partial Q^c_{[sd]}} = 0 \;\Rightarrow \phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^*}{\partial Q^c_{[s.]}} = 0 \;\Rightarrow \sum_{[.d]\in\mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c4$$

□

Theorem : <u>*sufficient conditions*</u>

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f*.*

Let $g(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *be continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi})\ \ \forall\ t \in [t_0, t_f]$.

Let $(\bar{\mathbf{X}}(t), \bar{\mathbf{\Phi}}(t), \bar{\mathbf{\Psi}}(t)) \in (\mathbf{R}^n, \mathbf{RC}, \mathbf{LS})$ *is an admissible pair for the Optimal Control Problem and* $H(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}(t))$ *is convex in* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{RC}, \mathbf{LS})$, $\forall t \in [t_0, t_f]$. *If* $\exists\ \mathbf{P}(t) : [t_0, t_f] \rightarrow \mathbf{R}^n$ *continuous and piecewise continuously differentiable vector function, such that* $\forall\ t \in [t_0, t_f]$*:*

$$\dot{\bar{\mathbf{X}}}(t) = \mathbf{f}(t, \bar{\mathbf{X}}(t), \bar{\mathbf{\Phi}}(t), \bar{\mathbf{\Psi}}(t))$$

$$\bar{\mathbf{X}}(t_0) = \mathbf{X}_0$$

$$\left[\frac{\partial H}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t)\right] * \bar{\phi}^c_{o[sd]}(t) = 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\frac{\partial H}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t)\right] * \bar{\phi}^c_{\pi[sd]}(t) = 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c, c$$

$$\left[\frac{\partial H}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)\right] * \bar{\psi}^c_{[sd]}(t) = 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\frac{\partial H}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t, \bar{\mathbf{X}}, \bar{\mathbf{\Phi}}(t), \bar{\mathbf{\Psi}}(t), \mathbf{P}(t))$$

$$\mathbf{P}(\mathbf{t}_f) = \mathbf{0}$$

$$\bar{\phi}^c_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \bar{\phi}^c_{\pi[sd]}(t) = 1 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d]\in\mathbf{D}^c_{[s.]}} \bar{\psi}^c_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\bar{\phi}^c_{o[sd]}(t),\ \bar{\phi}^c_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\bar{\psi}^c_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

then $(\bar{\mathbf{X}}(t), \bar{\mathbf{\Phi}}(t), \bar{\mathbf{\Psi}}(t))$ *is optimal.*

Proof: The proof is similar to that of the necessary conditions. In addition, we use the convexity of the Hamiltonian with respect to the state and controls.□

Theorem :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f.

Let $g(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *be continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{RC}, \mathbf{LS})$, $\forall\ t \in [t_0, t_f]$.

If $(\hat{\mathbf{\Phi}}^*(t, \mathbf{X}, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}, \mathbf{X}_0)) = (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a closed-loop memoryless team optimal solution such that* $(\hat{\mathbf{\Phi}}^*(t, \mathbf{X}, \mathbf{X}_0), \hat{\mathbf{\Phi}}^*(t, \mathbf{X}, \mathbf{X}_0))$ *is continuously differentiable with respect to* $\mathbf{X} \in \mathbf{R}^n$, $\forall\ c$, $t \in [t_0, t_f]$ *and* $\{\mathbf{X}^*(t),\ t \in [t_0, t_f]\}$ *is the corresponding state trajectory, then* $\exists\ \mathbf{P}(t) : [t_0, t_f] \rightarrow \mathbf{R}^n$, *continuous and piecewise continuously differentiable vector functions, such that* $\forall\ t \in [t_0, t_f]$:

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\left[\frac{\partial H^*}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{o[sd]}(t) = 0 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\frac{\partial H^*}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{\pi[sd]}(t) = 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c, c$$

$$\left[\frac{\partial H^*}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)\right] * \psi^{c*}_{[sd]}(t) = 0 \quad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\frac{\partial H^*}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t) \geq 0 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^*}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^*}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t, \mathbf{X}^*, \hat{\mathbf{\Phi}}^*(t, \mathbf{X}^*, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}^*, \mathbf{X}_0), \mathbf{P}(t))$$

$$\mathbf{P}\ (\mathrm{t}_f) = \mathbf{0}$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \Pi^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \Pi^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

Proof: The proof is similar to that of the open-loop solution.□

5.1.2 Dynamic Programming Formulation

In this section, we formulate the dynamic cooperative joint load sharing, routing and congestion control problem as a Dynamic Programming Problem (DPP). Algorithms for solving DPP's may be found in books by Bellman [31], Howard [220], Kumar & Varaiya [274] Bertsekas [37], Ross [406] among others.

Theorem :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f.

$(\mathbf{\Phi}^*, \mathbf{\Psi}^*) \in (\mathbf{RC}, \mathbf{LS})$ *is a team optimal solution if and only if the following conditions are satisfied:*

i) $\int_{t_0}^{t_f} g(t, \mathbf{X}^*(s), \hat{\mathbf{\Phi}}^*(\mathbf{X}^*(s))), \hat{\mathbf{\Psi}}^*(\mathbf{X}^*(s))ds = constant$

ii) $\exists\ \mathbf{X}^*,\ \mathbf{P}$ absolutely continuous such that:

$$\mathrm{H}\ (\mathrm{t}, \mathbf{X}^*(t), \mathbf{\Phi}^*(\mathbf{X}(t)), \mathbf{\Psi}^*(\mathbf{X}(t)), \mathbf{P}(t))\ -\ H^c(t, \mathbf{X}(t), \mathbf{\Phi}(\mathbf{X}(t)), \mathbf{\Psi}(\mathbf{X}(t)), \mathbf{P}(t))\ +$$

$$+\ \dot{\mathbf{P}}(t) * (\mathbf{X}^*(t) - \mathbf{X})\ \leq 0 \qquad a.e.\ \ t \in [t_0, t_f],\ \forall\ \mathbf{X} \in \mathbf{R}^n,\ (\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$$

$$\mathbf{P}\ (\mathrm{t}_f) * (\mathbf{X}^*(t_f) - \mathbf{X}) \leq 0 \quad \forall\ \mathbf{X} \in \mathbf{R}^n$$

Proof: Substituting the state equation in ii) and integrating it, we get the definition of the team-optimal solution. □

Definition :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f.

Under the memoryless perfect state or closed-loop perfect state information structure, $(\hat{\mathbf{\Phi}}, \hat{\mathbf{\Psi}}) \in (\mathbf{RC}, \mathbf{LS})$ *constitutes a* <u>*feedback team optimal solution*</u> *solution if and only if* $\exists\ V : [t_0, t_f] * \mathbf{R}^n \rightarrow R$ *satisfying the following relations:*

$$
\begin{aligned}
\mathrm{V(t,\mathbf{X})} &= \int_t^{t_f} g(s, \mathbf{X}^*(s), \hat{\mathbf{\Phi}}^*(s, \mathbf{I}(s)), \hat{\mathbf{\Psi}}^*(s, \mathbf{I}(s)))ds \leq \\
&\leq \int_t^{t_f} g(s, \mathbf{X}^*(s), \hat{\mathbf{\Phi}}(s, \mathbf{I}(s)), \hat{\mathbf{\Psi}}(s, \mathbf{I}(s)))ds \\
&\quad \forall\ (\hat{\mathbf{\Phi}}(s, \mathbf{I}(s)), \hat{\mathbf{\Psi}}(s, \mathbf{I}(s))) \in (\mathbf{RC}, \mathbf{LS}),\ \mathbf{X} \in \mathbf{R}^n
\end{aligned}
$$

$$
\begin{aligned}
&\quad \text{such that} \quad \forall\ s \in [t, t_f] \\
\dot{\mathbf{X}}(s) &= \mathrm{f}(\mathrm{s}, \mathbf{X}\ (\mathrm{s}), \hat{\mathbf{\Phi}}(s, \mathbf{I}(s)), \hat{\mathbf{\Psi}}(s, \mathbf{I}(s))) \\
\mathbf{X}\ (\mathrm{t}) &= \mathbf{X} \\
\dot{\mathbf{X}}^*(s) &= \mathrm{f}(\mathrm{s}, \mathbf{X}^*(s), \hat{\mathbf{\Phi}}^*(s, \mathbf{I}(s)), \hat{\mathbf{\Psi}}^*(s, \mathbf{I}(s))) \\
\mathbf{X}^*(s) &= \mathbf{X}
\end{aligned}
$$

where $\mathbf{I}(s) = \{\mathbf{X}(s), \mathbf{X}_0\}$ or $\mathbf{I}(s) = \{\mathbf{X}(\tau),\ \tau \leq s\}$.

$V(t, \mathbf{X})$ is the value function associated with the optimal control problem of minimizing J over $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Psi}}) \in \mathbf{LS}, \mathbf{RC})$.

The concept of feedback team optimal solution means that if $(\mathbf{\Phi}(s), \mathbf{\Psi}(s))$ is a feedback team optimal solution to the problem during $[t_0, t_f]$, is also a feedback team optimal solution to the problem during $[t, t_f]$, with the initial state taken as $\mathbf{X}(t)$. So, feedback team optimal solution strategies will depend only on the time variable and the current value of the state, but not on memory.

Proposition :

Every open-loop team optimal solution for the dynamic joint load sharing, routing and congestion control problem among cooperative classes is also closed-loop team optimal solution.

Proposition :

Under the memoryless (respectively, closed-loop) perfect state information structure, every feedback team optimal solution for the dynamic joint load sharing, routing and congestion control problem among cooperative classes is a closed-loop no memory (respectively, closed-loop) team optimal solution.

Theorem :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 and final time t_f.

Under the memory perfect state or closed loop perfect state information structure, $(\hat{\mathbf{\Phi}}, \hat{\mathbf{\Psi}}) \in (\mathbf{RC}, \mathbf{LS})$ provides a feedback team optimal solution if $\exists V : [t_0, t_f] * \mathbf{R}^n \rightarrow \mathbf{R}$ continuously differentiable satisfying the partial differential equations

$$-\frac{\partial V(t,\mathbf{X})}{\partial t} = \min_{(\mathbf{\Phi},\mathbf{\Psi}) \in (\mathbf{RC},\mathbf{LS})} \left\{ \frac{\partial V(t,\mathbf{X})}{\partial \mathbf{X}} * \mathbf{f}(t,\mathbf{X},\mathbf{\Phi},\mathbf{\Psi}) + g(t,\mathbf{X},\mathbf{\Phi},\mathbf{\Psi}) \right\} =$$

$$= \frac{\partial V(t,\mathbf{X})}{\partial \mathbf{X}} * \mathbf{f}(t,\mathbf{X},\hat{\mathbf{\Phi}}^*(t,\mathbf{X}),\hat{\mathbf{\Psi}}^*(t,\mathbf{X})) + g(t,\mathbf{X},\hat{\mathbf{\Phi}}^*(t,\mathbf{X}),\hat{\mathbf{\Psi}}^*(t,\mathbf{X}))$$

The above equation is called Hamilton-Jacobi-Bellman (H-J-B) equation.

5.1.3 Nonlinear Complementarity Problem Formulation

In this section, we formulate the dynamic cooperative load sharing, routing and congestion control problem as a Nonlinear Complementarity Problem (NCP).

Define the vector of class load sharing, routing and load sharing fractions as well as Lagrange multipliers:

$$\mathbf{Z}(t) = [\ldots\ \phi^c_{o[sd]}(t)\ \ldots\ \phi^c_{\pi[sd]}\ \ldots\ Q^c_{[sd]}(t)\ \ldots.\ \psi^c_{[sd]}(t)\ \ldots\ Q^c_{[s.]}(t)\ \ldots]^T$$

and the vector of class derivative of the Lagrangian with respect to the congestion control, routing and load sharing fractions as well as Lagrange multipliers:

$$\nabla L(t,\mathbf{X}(t),\mathbf{Z}(t)) = \Bigg[\dots \left(\frac{\partial H}{\partial \phi^c_{o[sd]}} - Q^c_{o[sd]}(t)\right) \dots \left(\frac{\partial H}{\partial \phi^c_{\pi[sd]}} - Q^c_{\pi[sd]}(t)\right) \dots$$

$$\dots \left(1 - \phi^c_{o[sd]}(t) - \sum_{\pi[sd]\in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}(t)\right) \dots$$

$$\dots \left(\frac{\partial H}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)\right) \dots \left(1 - \sum_{[.d]\in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]}(t)\right) \dots \Bigg]$$

Theorem :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f*.*

Let $g(t,\mathbf{X},\mathbf{\Phi},\mathbf{\Psi})$, $\mathbf{f}(t,\mathbf{X},\mathbf{\Phi},\mathbf{\Psi})$, *be continuously differentiable with respect to* $(\mathbf{X},\mathbf{\Phi},\mathbf{\Psi}) \in (\mathbf{R}^n,\mathbf{\Phi},\mathbf{\Psi})$ $\forall\ t \in [t_0,t_f]$. *If* H *is differentiable and convex in* $(\mathbf{X},\mathbf{\Phi},\mathbf{\Psi}) \in (\mathbf{RC},\mathbf{LS})$, $\forall\ t \in [t_0,t_f]$,

then $(\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t)) \in (\mathbf{RC},\mathbf{LS})$ *is a team optimal solution if and only if it solves the following* <u>*Nonlinear Complementarity Problem*</u> $\forall\ t \in [t_0,t_f]$:

$\nabla L(t,\mathbf{X}^*(t),\mathbf{Z}^*(t)) * \mathbf{Z}^*(t) = 0$

$\nabla L(t,\mathbf{X}^*(t),\mathbf{Z}^*(t)) \geq 0$

$\mathbf{Z}^*(t) \geq 0$

$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t,\mathbf{X}^*(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))$

$\mathbf{X}^*(t_0) = \mathbf{X}_0$

$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t,\mathbf{X}^*,\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t),\mathbf{P}(t))$

$\mathbf{P}(\mathrm{t}_f) = \mathbf{0}$

Proof: After some algebraic manipulations, we find that the NCP: $\nabla L(\mathbf{Z}(t)) * \mathbf{Z}(t) = 0$ $\nabla L(\mathbf{Z}(t)) \geq 0$ $\mathbf{Z}(t) \geq 0$ with $\mathbf{Z}(t)$ and $\nabla L(\mathbf{Z}(t))$ as defined above, is equivalent to the Pontryagin's maximum principle necessary conditions. □

5.1.4 Variational Inequality Formulation

In this section, we formulate the dynamic cooperative load sharing, routing and congestion control problem as a Variational Inequality Problem (VIP).

Define the vector of class congestion control, routing and load sharing fractions:

$$(\mathbf{\Phi}(t), \mathbf{\Psi}(t)) = \left[\dots\ \phi^c_{o[sd]}(t)\ \dots\ \phi^c_{\pi[sd]}(t)\ \dots\ \psi^c_{[sd]}(t)\ \dots \right]^T$$

as well the vector of class derivatives of the cost function with respect to the congestion control, routing and load sharing fractions:

$$\nabla H(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t), \mathbf{P}(t)) \quad = \quad \left[\dots \frac{\partial H}{\partial \phi^c_{o[sd]}} \dots \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial H}{\partial \phi^c_{\pi[sd]}} \dots \frac{\partial H}{\partial \psi^c_{[sd]}} \dots \right]$$

Theorem :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f.

Let $g(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *be continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi})\ \ \forall\ t \in [t_0, t_f]$. *If* H *is continuously differentiable and convex in* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{RC}, \mathbf{LS})$, $\ \forall\ t \in [t_0, t_f]$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a team optimal solution if and only if it solves the following Variational Inequality Problem* $\forall\ t \in [t_0, t_f]$:

$$\nabla H(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t)) * ((\mathbf{\Phi}, \mathbf{\Psi}) - (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))) \geq 0$$
$$\forall\ (\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$$
$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$
$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$
$$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))$$
$$\mathbf{P}(t_f) = \mathbf{0}$$

Proof: If $(\mathbf{\Phi}(t), \mathbf{\Psi}(t))$ is a local minimum for the following minimization problem

$$
\begin{array}{ll}
\textit{minimize} & \int_{t_0}^{t_f} g(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))dt \\
\textit{with respect to} & (\Phi(t), \mathbf{\Psi}(t)) \\
\textit{such that} & \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\
& \mathbf{X}(\mathrm{t}_0) = \mathbf{X}_0 \\
& \\
& (\Phi(t), \mathbf{\Psi}(t)) \in (\mathbf{RC}, \mathbf{LS})
\end{array}
$$

and g is a continuously differentiable convex function over the nonempty convex, closed and bounded set $(\mathbf{RC}, \mathbf{LS})$, then $\forall\ t \in [t_0, t_f]$:

$$
\sum_{c} \sum_{[sd] \in \mathbf{SD}^c} \left\{ \frac{\partial H^*}{\partial \phi^c_{o[sd]}} * (\phi^c_{o[sd]} - \phi^{c*}_{o[sd]}(t)) + \right.
$$

$$
+ \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial H^*}{\partial \phi^c_{\pi[sd]}} * (\phi^c_{\pi[sd]} - \phi^{c*}_{\pi[sd]}(t)) +
$$

$$
\left. + \frac{\partial H^*}{\partial \psi^c_{[sd]}} * (\psi^c_{[sd]} - \psi^{c*}_{[sd]}(t)) \right\} \geq 0 \quad \forall\ (\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})
$$

□

Another equivalent VIP formulation is the following Theorem:

Theorem :

Consider the dynamic joint load sharing, routing and congestion control problem in distributed systems with multiple cooperative classes, with fixed initial time t_0 *and final time* t_f.

Let $g(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *be continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi}) \quad \forall\ t \in [t_0, t_f]$. *If* H *is continuously differentiable and convex in* $(\mathbf{H}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{RC}, \mathbf{LS}), \quad \forall\ t \in [t_0, t_f]$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a team optimal solution if and only if it solves the following* *Variational Inequality Problem* $\forall\ t \in [t_0, t_f]$:

$$\nabla L(t, \mathbf{X}^*(t), \mathbf{Z}^*(t)) * (\mathbf{Z} - \mathbf{Z}^*(t)) \geq 0 \qquad \forall\ \mathbf{Z} > 0$$

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t))$$

$$\mathbf{P}(\mathrm{t}_f) = \mathbf{0}$$

The NCP: $f(x^*) * x^* = 0 \quad f(x^*) \geq 0 \quad x^* > 0$

and the VIP: find x^* such that $f(x^*) * (x - x^*) \geq 0 \quad \forall x > 0$

are equivalent.□

5.1.5 Maximum Principle for Separable Cost Functions

In this section, we derive the first order necessary conditions for a team optimal solution on the path flows, when the cost function of each resource depends only on the flow on this resource.

According to the team optimal solution definition, each class c minimizes its cost function g given the optimum decisions of all other classes.

$$\textit{minimize} \qquad \int_{t_0}^{t_f} g(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))dt =$$

$$= \sum_{ij} \int_{t_0}^{t_f} g_{ij}(t, \mathbf{X}_{ij}(t), \mathbf{\Lambda}_{ij}(t)_d t \ +$$

$$+ \ \sum_{i} \int_{t_0}^{t_f} g_i(t, \mathbf{X}_i(t), \mathbf{\Lambda}_i(t))dt \ +$$

$$+ \ \sum_{[sd]} \int_{t_0}^{t_f} g_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Lambda}_{o[sd]}(t))dt \ +$$

$$+ \ \sum_{[.d]} \int_{t_0}^{t_f} g_{[.d]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Lambda}_{[.d]}(t))dt$$

$$\textit{with respect to} \qquad (\Phi(t), \mathbf{\Psi}(t))$$

such that

$$
\begin{aligned}
\dot{\mathbf{X}}^k_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
E\mathbf{X}^k_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}
$$

$$\phi^c_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c$$

$$\phi^c_{o[sd]}(t),\ \phi^c_{\pi[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c$$

$$\psi^c_{[sd]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c$$

Pontryagin's maximum principle necessary conditions are $\forall\ t \in [t_0, t_f]$:

$$
\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^{k}_{ij[sd]}(t, \mathbf{X}^{*}_{ij}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^{k}_{i[sd]}(t, \mathbf{X}^{*}_{i}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^{k}_{o[sd]}(t, \mathbf{X}^{*}_{o[sd]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^{k}_{[.d][sd]}(t, \mathbf{X}^{*}_{[.d]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^{k}_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^{k}_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^{k}_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^{k}_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^{k},\ k
\end{aligned}
$$

$$\left[\frac{\partial g_{[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))}{\partial\phi^c_{o[sd]}}+\right.$$

$$+\sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}}\mathbf{f}^k_{o[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))-$$

$$\left.-Q^c_{[sd]}(t)\right] * \phi^{c*}_{o[sd]}(t) = 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\sum_{ij}\frac{\partial g_{ij}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))}{\partial\phi^c_{\pi[sd]}}+\sum_i\frac{\partial g_i(t,\mathbf{X}^*_i(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))}{\partial\phi^c_{\pi[sd]}}+\right.$$

$$+\sum_k\sum_{[s'd']}\sum_{ij}\mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}}\mathbf{f}^k_{ij[s'd']}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+$$

$$+\sum_k\sum_{[s'd']}\sum_{i}\mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}}\mathbf{f}^c_{i[s'd']}(t,\mathbf{X}^*_{i}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))-$$

$$\left.-Q^c_{[sd]}(t)\right] * \phi^{c*}_{\pi[sd]}(t) = 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$
\left[\sum_{ij} \frac{\partial g_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} + \sum_{i} \frac{\partial g_{i}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} + \right.
$$

$$
+ \frac{\partial g_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} + \frac{\partial g_{[.d]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} +
$$

$$
+ \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +
$$

$$
+ \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{i[s'd']}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +
$$

$$
+ \sum_{k} \mathbf{P}^{k,c}_{o[sd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +
$$

$$
+ \sum_{k} \sum_{[s'.]} \mathbf{P}^{k,c}_{[.d][s'd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -
$$

$$
\left. - Q^c_{[s.]}(t) \right] * \psi^{c*}_{[sd]}(t) = 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c
$$

$$\frac{\partial g_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{o[sd]}} +$$

$$\sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\text{-}Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \frac{\partial g_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} + \sum_i \frac{\partial g_i(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} +$$

$$+ \sum_k \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_k \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\text{-}Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \frac{\partial g_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} + \sum_{i} \frac{\partial g_{i}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} +$$

$$+ \frac{\partial g_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} + \frac{\partial g_{[.d]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} +$$

$$+ \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{i[s'd']}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \mathbf{P}^{k,c}_{o[sd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \sum_{[s'.]} \mathbf{P}^{k,c}_{[.d][s'd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-Q^c_{[s.]}(t) \geq 0 \qquad \forall\, [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{i[s'd']}(t) * \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \mathbf{P}^{c,n}_{o[sd]}(t) * \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

The partial derivatives of the cost function $g(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$ with respect to the path tractions $\phi^c_{\pi[sd]}$ can be written with respect to the link flows λ^c_{ij} and node flows λ^c_i:

$$\frac{\partial g_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^c_{\pi[sd]}} = \frac{\partial g_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda^c_{ij}} * \frac{\partial \lambda^c_{ij}}{\partial \phi^c_{\pi[sd]}} =$$

$$= \frac{\partial g_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda^c_{ij}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}) * 1_{ij \in \pi[sd]}(t)$$

$$\frac{\partial g_i(t, \mathbf{X}_i, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^c_{\pi[sd]}} = \frac{\partial g_i(t, \mathbf{X}_i, \mathbf{\Lambda}_i)}{\partial \lambda^c_i} * \frac{\partial \lambda^c_i}{\partial \phi^c_{\pi[sd]}} =$$

$$= \frac{\partial g_i(t, \mathbf{X}_i, \mathbf{\Lambda}_i)}{\partial \lambda^c_i} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}) * 1_{i \in \pi[sd]}(t)$$

$$\frac{\partial g_{[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^c_{o[sd]}} = \frac{\partial g_{[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda^c_{o[sd]}} * \frac{\partial \lambda^c_{o[sd]}}{\partial \phi^c_{o[sd]}} =$$

$$= \frac{\partial g_{[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]})$$

$$\frac{\partial g_{ij}(t,\mathbf{X}_{ij},\mathbf{\Phi},\mathbf{\Psi})}{\partial \psi^c_{[sd]}} = \frac{\partial g_{ij}(t,\mathbf{X}_{ij},\mathbf{\Lambda}_{ij})}{\partial \lambda^c_{ij}} * \frac{\partial \lambda^c_{ij}}{\partial \psi^c_{[sd]}} =$$

$$= \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \frac{\partial g_{ij}(t,\mathbf{X}_{ij},\mathbf{\Lambda}_{ij})}{\partial \lambda^c_{ij}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{\pi[sd]} * 1_{ij\in\pi[sd]}(t)$$

$$\frac{\partial g_{i}(t,\mathbf{X}_{i},\mathbf{\Phi},\mathbf{\Psi})}{\partial \psi^c_{[sd]}} = \frac{\partial g_{i}(t,\mathbf{X}_{i},\mathbf{\Lambda}_{i})}{\partial \lambda^c_{i}} * \frac{\partial \lambda^c_{i}}{\partial \psi^c_{[sd]}} =$$

$$= \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \frac{\partial g_{i}(t,\mathbf{X}_{i},\mathbf{\Lambda}_{i})}{\partial \lambda^c_{i}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{\pi[sd]} * 1_{i\in\pi[sd]}(t)$$

$$\frac{\partial g_{[sd]}(t,\mathbf{X}_{o[sd]},\mathbf{\Phi},\mathbf{\Psi})}{\partial \psi^c_{[sd]}} = \frac{\partial g_{[sd]}(t,\mathbf{X}_{o[sd]},\mathbf{\Lambda}_{o[sd]})}{\partial \lambda^c_{o[sd]}} * \frac{\partial \lambda^c_{o[sd]}}{\partial \psi^c_{[sd]}} =$$

$$= \frac{\partial g_{[sd]}(t,\mathbf{X}_{o[sd]},\mathbf{\Lambda}_{o[sd]})}{\partial \lambda^c_{o[sd]}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{o[sd]}$$

$$\frac{\partial g_{[.d]}(t,\mathbf{X}_{[.d]},\mathbf{\Phi},\mathbf{\Psi})}{\partial \psi^c_{[sd]}} = \frac{\partial g_{[.d]}(t,\mathbf{X}_{[.d]},\mathbf{\Lambda}_{[.d]})}{\partial \lambda^c_{[.d]}} * \frac{\partial \lambda^c_{[.d]}}{\partial \psi^c_{[sd]}} =$$

$$= \frac{\partial g_{[.d]}(t,\mathbf{X}_{[.d]},\mathbf{\Lambda}_{[.d]})}{\partial \lambda^c_{[.d]}} * \gamma^c_{[s.]}(t)$$

Then Pontryagin's maximum principle becomes $\forall\ t \in [t_0, t_f]$:

$$
\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^{k}_{ij[sd]}(t, \mathbf{X}^{*}_{ij}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^{k}_{i[sd]}(t, \mathbf{X}^{*}_{i}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall \subset,\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^{k}_{o[sd]}(t, \mathbf{X}^{*}_{o[sd]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^{k}_{[.d][sd]}(t, \mathbf{X}^{*}_{[.d]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^{k}_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^{k}_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^{k}_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^{k}_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^{k},\ k
\end{aligned}
$$

$$\left[\frac{\partial g_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Lambda}^*_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) + \right.$$

$$+ \sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\left. - Q^c_{[sd]}(t) \right] * \phi^{c*}_{o[sd]}(t) = 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\sum_{ij} \frac{\partial g_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Lambda}^*_{ij}(t))}{\partial \lambda^c_{ij}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{ij \in \pi[sd]}(t) + \right.$$

$$+ \sum_i \frac{\partial g_i(t, \mathbf{X}^*_i(t), \mathbf{\Lambda}^*_i(t))}{\partial \lambda^c_i} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{i \in \pi[sd]}(t) +$$

$$+ \sum_k \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_k \sum_{[s'd']} \sum_i \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\left. - Q^c_{[sd]}(t) \right] * \phi^{c*}_{\pi[sd]}(t) = 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$
\left[\sum_{ij}\sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}}\frac{\partial g_{ij}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Lambda}^*_{ij}(t))}{\partial\lambda^c_{ij}}*\gamma^c_{[s.]}(t)*\phi^{c*}_{\pi[sd]}(t))*1_{ij\in\pi[sd]}(t)+\right.
$$

$$
+\sum_{i}\sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}}\frac{\partial g_{i}(t,\mathbf{X}^*_{i}(t),\mathbf{\Lambda}^*_{i}(t))}{\partial\lambda^c_{i}}*\gamma^c_{[s.]}(t)*\phi^{c*}_{\pi[sd]}(t))*1_{i\in\pi[sd]}(t)+
$$

$$
+\frac{\partial g_{[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Lambda}^*_{o[sd]}(t))}{\partial\lambda^c_{o[sd]}}*\gamma^c_{[s.]}(t)*\phi^{c*}_{o[sd]}(t))+
$$

$$
+\frac{\partial g_{[.d]}(t,\mathbf{X}^*_{[.d]}(t),\mathbf{\Lambda}^*_{[.d]}(t))}{\partial\lambda^c_{[.d]}}*\gamma^c_{[s.]}(t)+
$$

$$
+\sum_{k}\sum_{[s'd']}\sum_{ij}\mathbf{P}^{k,c}_{ij[s'd']}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{ij[s'd']}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+
$$

$$
+\sum_{k}\sum_{[s'd']}\sum_{i}\mathbf{P}^{k,c}_{i[s'd']}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{i[s'd']}(t,\mathbf{X}^*_{i}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+
$$

$$
+\sum_{k}\mathbf{P}^{k,c}_{o[sd]}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{o[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+
$$

$$
+\sum_{k}\sum_{[s'.]}\mathbf{P}^{k,c}_{[.d][s'd]}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{[.d][s'd]}(t,\mathbf{X}^*_{[.d]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))-
$$

$$
\left.-Q^c_{[s.]}(t)\right]*\psi^{c*}_{[sd]}(t)=0\qquad\forall\ [.d]\in\mathbf{D}^c_{[s.]},\ [s.]\in\mathbf{S}^c,\ c
$$

$$\frac{\partial g_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Lambda}^*_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) +$$

$$+ \sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-Q^c_{[sd]}(t) \geq 0 \qquad\qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \frac{\partial g_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Lambda}^*_{ij}(t))}{\partial \lambda^c_{ij}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{ij \in \pi[sd]}(t) +$$

$$+ \sum_i \frac{\partial g_i(t, \mathbf{X}^*_i(t), \mathbf{\Lambda}^*_i(t))}{\partial \lambda^c_i} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{i \in \pi[sd]}(t) +$$

$$+ \sum_k \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_k \sum_{[s'd']} \sum_i \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-Q^c_{[sd]}(t) \geq 0 \qquad\qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \frac{\partial g_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Lambda}^*_{ij}(t))}{\partial \lambda^c_{ij}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{\pi[sd]}(t)) * 1_{ij\in\pi[sd]}(t)+$$

$$+\sum_{i} \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \frac{\partial g_{i}(t, \mathbf{X}^*_{i}(t), \mathbf{\Lambda}^*_{i}(t))}{\partial \lambda^c_{i}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{\pi[sd]}(t)) * 1_{i\in\pi[sd]}(t)+$$

$$+\frac{\partial g_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Lambda}^*_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{o[sd]}(t))+$$

$$+\frac{\partial g_{[.d]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Lambda}^*_{[.d]}(t))}{\partial \lambda^c_{[.d]}} * \gamma^c_{[s.]}(t)+$$

$$+\sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))+$$

$$+\sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{i[s'd']}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))+$$

$$+\sum_{k} \mathbf{P}^{k,c}_{o[sd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))+$$

$$+\sum_{k} \sum_{[s'.]} \mathbf{P}^{k,c}_{[.d][s'd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))-$$

$$-Q^c_{[s.]}(t) \geq 0 \qquad\qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{i[s'd']}(t) * \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \mathbf{P}^{c,n}_{o[sd]}(t) * \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

Next, for each class c, we define the length for the rejected flow $[sd]$, the length for the path $\pi[sd]$ and the length for the source-destination pair $[sd]$:

$$l^{c,team}_{o[sd]}(t) = \frac{\partial g_{[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Lambda}_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}(t)) +$$

$$+ \sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$$

$$\forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$
\begin{aligned}
l_{\pi[sd]}^{c,team}(t) = & \sum_{ij} \frac{\partial g_{ij}(t, \mathbf{X}_{ij}(t), \mathbf{\Lambda}_{ij}(t))}{\partial \lambda_{ij}^{c}} * (\gamma_{[sd]}^{c}(t) + \gamma_{[s.]}^{c}(t) * \psi_{[sd]}^{c}(t)) * 1_{ij\in\pi[sd]}(t) + \\
& + \sum_{i} \frac{\partial g_{i}(t, \mathbf{X}_{i}(t), \mathbf{\Lambda}_{i}(t))}{\partial \lambda_{i}^{c}} * (\gamma_{[sd]}^{c}(t) + \gamma_{[s.]}^{c}(t) * \psi_{[sd]}^{c}(t)) * 1_{i\in\pi[sd]}(t) + \\
& + \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}_{ij[s'd']}^{k,c}(t) * \nabla_{\phi_{\pi[sd]}^{c}} \mathbf{f}_{ij[s'd']}^{k}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) + \\
& + \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}_{i[s'd']}^{k,c}(t) * \nabla_{\phi_{\pi[sd]}^{c}} \mathbf{f}_{i[s'd']}^{c}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))
\end{aligned}
$$

$$
\forall\ \pi[sd] \in \mathbf{\Pi}_{[sd]}^{c},\ [sd] \in \mathbf{SD}^{c},\ c
$$

$$
\begin{aligned}
l^{c,team}_{[sd]}(t) = & \sum_{ij} \sum_{\pi[sd]\in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g_{ij}(t, \mathbf{X}_{ij}(t), \mathbf{\Lambda}_{ij}(t))}{\partial \lambda^c_{ij}} * \gamma^c_{[s.]}(t) * \phi^c_{\pi[sd]}(t)) * 1_{ij\in\pi[sd]}(t)+ \\
& + \sum_{i} \sum_{\pi[sd]\in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g_{i}(t, \mathbf{X}_{i}(t), \mathbf{\Lambda}_{i}(t))}{\partial \lambda^c_{i}} * \gamma^c_{[s.]}(t) * \phi^c_{\pi[sd]}(t)) * 1_{i\in\pi[sd]}(t)+ \\
& + \frac{\partial g_{[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Lambda}_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * \gamma^c_{[s.]}(t) * \phi^c_{o[sd]}(t))+ \\
& + \frac{\partial g_{[.d]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Lambda}_{[.d]}(t))}{\partial \lambda^c_{[.d]}} * \gamma^c_{[s.]}(t)+ \\
& + \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))+ \\
& + \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{i[s'd']}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))+ \\
& + \sum_{k} \mathbf{P}^{k,c}_{o[sd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))+ \\
& + \sum_{k} \sum_{[s'.]} \mathbf{P}^{k,c}_{[.d][s'd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{[.d][s'd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))-
\end{aligned}
$$

$$
\forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c
$$

External arriving flow at a source is assigned to the destination that has the minimum length from the source. However, this flow may be rejected if the length of rejecting it is less than the lengths of the paths to its destination. If it is accepted, then it is routed to its destination via the minimum length path.

In the next section, we will derive the same conditions by an alternative way, and we shall state the above ideas more formally.

5.1.6 V.I. for Separable Cost Functions

Equivalently, the team optimal solution definition, each class c minimizes its cost function g given the optimum decisions of all other classes. We first solve the routing and congestion control problem assuming that all other classes act optimally for themselves. So, class c first solves the routing and congestion control problems

$$
\begin{aligned}
\textit{minimize} \qquad & \int_{t_0}^{t_f} g(t, \mathbf{X}, \mathbf{\Phi}(t), \mathbf{\Psi}(t))dt = \\
& = \sum_{ij} \int_{t_0}^{t_f} g_{ij}(t, \mathbf{X}_{ij}(t), \mathbf{\Lambda}_{ij}(t))dt \ + \\
& + \ \sum_{i} \int_{t_0}^{t_f} g_{i}(t, \mathbf{X}_{i}(t), \mathbf{\Lambda}_{i}(t))dt \ + \\
& + \ \sum_{[sd]} \int_{t_0}^{t_f} g_{[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Lambda}_{o[sd]}(t))dt \ + \\
& + \ \sum_{[.d]} \int_{t_0}^{t_f} g_{[.d]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Lambda}_{[.d]}(t))dt \\
\textit{with respect to} \qquad & \mathbf{\Phi}(t)
\end{aligned}
$$

such that

$$\begin{aligned}
\dot{\mathbf{X}}^k_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\phi^c_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c$$

$$\phi^c_{o[sd]}(t),\ \phi^c_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c$$

The necessary optimality conditions are $\forall\ t \in [t_0, t_f]$:

$$\sum \sum_{[sd]\in \mathbf{SD}^c} \left\{ \frac{\partial g(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{o[sd]} * (\phi^c_{o[sd]} - \phi^{c*}_{o[sd]}(t))} + \right.$$

$$\left. + \sum_{\pi[sd]\in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]} * (\phi^c_{\pi[sd]} - \phi^{c*}_{\pi[sd]}(t))} \right\} \geq 0 \qquad \forall\ \mathbf{\Phi}^c \in \mathbf{RC}^c$$

such that

$$\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) = \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{i[sd]}(t) = \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{o[sd]}(t) = \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) = \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{ij[sd]}(t_0) = \mathbf{X}^k_{ij[sd],0} \qquad \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{i[sd]}(t_0) = \mathbf{X}^k_{i[sd],0} \qquad \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{o[sd]}(t_0) = \mathbf{X}^k_{o[sd],0} \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{[.d][sd]}(t_0) = \mathbf{X}^k_{[.d][sd],0} \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g_i(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_o, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c$$

We can decompose these conditions for each source-destination pair $[sd] \in \mathbf{SD}^c$ $\forall\, t \in [t_0, t_f]$:

$$\frac{\partial g(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{o[sd]}} * (\phi^c_{o[sd]} - \phi^{c*}_{o[sd]}(t)) +$$

$$+ \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} * (\phi^c_{\pi[sd]} - \phi^{c*}_{\pi[sd]}(t)) \geq 0 \qquad \forall\ \mathbf{\Phi}^c \in \mathbf{RC}^c$$

such that

$$\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ ij,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g_i(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_o, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]} = 1$$

$$\phi^c_{o[sd]}(t),\ \phi^c_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]}$$

Theorem : <u>*Routing*</u>

$$\phi^{c*}_{\pi[sd]}(t) > 0 \quad only \quad if \qquad l^{c,team*}_{\pi[sd]}(t) = \min\{l^{c,team*}_{o[sd]}(t), \min_{p[sd]}\{l^{c,team*}_{p[sd]}(t)\}\}$$

$$\phi^{c*}_{\pi[sd]}(t) = 0 \qquad o.w.$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^{c}_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1$$

$$\forall\ \pi[sd] \in \Pi^{c}_{[sd]}, \quad [sd] \in \mathbf{SD}^{c}, \quad c$$

and satisfies the partial differential vectors for the state and the costate variables.

Theorem : <u>*Congestion Control*</u>

Flow is not admitted into the network only if its rejection length is less than the minimum length path to its destination:

$$\phi^{c*}_{o[sd]}(t) > 0 \quad only \quad if \qquad l^{c,team*}_{o[sd]}(t) = \min\{l^{c,team*}_{o[sd]}(t), \min_{p[sd]}\{l^{c,team*}_{p[sd]}(t)\}\}$$

$$\phi^{c*}_{\pi[sd]} = 0 \qquad o.w.$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^{c}_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^{c}, \quad c$$

and satisfies the partial differential vectors for the state and the costate variables.

Having found the optimum routing and congestion control decisions, we proceed to solve the load sharing problem for class c assuming also that all other classes act at their optimum decisions. So, the load sharing problem for class c is

$$\textit{minimize} \qquad \int_{t_0}^{t_f} g(t, \mathbf{X}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}(t))dt =$$

$$= \sum_{ij} \int_{t_0}^{t_f} g_{ij}(t, \mathbf{X}_{ij}(t), \mathbf{\Lambda}_{ij}(t))dt \; +$$

$$+ \; \sum_{i} \int_{t_0}^{t_f} g_i(t, \mathbf{X}_i(t), \mathbf{\Lambda}_i(t))dt \; +$$

$$+ \; \sum_{[sd]} \int_{t_0}^{t_f} g_{[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Lambda}_{o[sd]}(t))dt \; +$$

$$+ \; \sum_{[.d]} \int_{t_0}^{t_f} g_{[.d]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Lambda}_{[.d]}(t))dt$$

with respect to $\mathbf{\Psi}(t)$

such that

$$\begin{aligned}
\dot{\mathbf{X}}^k_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c$$

$$\psi^c_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c$$

The necessary and sufficient optimality conditions are $\forall\ t \in [t_0, t_f]$:

$$\sum_c \frac{\partial g(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} * (\psi^c_{[sd]} - \psi^{c*}_{[sd]}(t)) \quad \geq 0 \quad \forall\ \mathbf{\Psi} \in \mathbf{LS}$$

such

$$\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g_i(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_o, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c$$

We can decompose these conditions for each source node $[s.] \in \mathbf{S}^c \ \ \forall\, t \in [t_0, t_f]$:

$$\sum_{[.d]\in \mathbf{D}^c_{[s.]}} \frac{\partial g(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} * (\psi^c_{[sd]} - \psi^{c*}_{[sd]}(t)) \quad \geq 0 \qquad \forall\ \mathbf{\Psi}^c \in \mathbf{LS}^c$$

such that

$$\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) & \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) & \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) & \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) & \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} & \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} & \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} & \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} & \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g_i(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_i, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_o, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]}$$

Theorem : Load Sharing

For each source, there must be flow only to destinations whose length is minimum:

$$\psi^{c*}_{[sd]}(t) > 0 \quad only \;\; if \qquad l^{c,team*}_{[sd]}(t) = \min_{[sd']}\{l^{c,team*}_{[sd']}(t)\}$$

$$\psi^{c*}_{[sd]}(t) = 0 \qquad\qquad\qquad o.w.$$

$$\sum_{[.d]\in \mathbf{S}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \;\; \forall\, [.d] \in \mathbf{D}^c_{[s.]},\; [s.] \in \mathbf{S}^c,\; c$$

and satisfies the partial differential vectors for the state and the costate variables.

Thus, in this section we have formulated and solved the load sharing, routing and congestion control problem as a team problem among multiple cooperative classes.

5.2 Nash Equilibrium Solution

In this section, we formulate the dynamic join load sharing, routing and congestion control problem on the path flow space as a non-cooperative dynamic Nash game among competing classes.

Customers of each class try to use the resources of the distributed system for their own benefit, ignoring the inconvenience that they cause to customers from other classes. Since the behavior of each class is similar to that of any other class, i.e. to operate optimally for its customers, next we consider customers only from class c, and the effect of customers from other classes on them.

After the static non-cooperative games by Nash [347], dynamic non-cooperative games have been investigated and are presented in books by: Isaacs [231] Blaquiete, Gerard & Leitmann [55] Friedman [173] Case [87], Rosenmuller [404], Mehlmann [328], Krasovskii & Subbotin [265] among others.

Next, we briefly review research on dynamic Nash games:

Berkovitz [34] obtains necessary conditions for zero-sum differential games. Sarma, Ragade & Prasad [425] introduce dynamic n-person noncooperative dynamic games and provide necessary conditions. Case [88] provides sufficient conditions and use dynamic programming arguments. Stalford & Leitmann [459] discuss sufficiency conditions for dynamic Nash games.

Sandell [417] proves that for deterministic nonzero-sum games any open-loop Nash strategy is also a closed-loop strategy. Williams [513] obtains sufficient conditions for the existence of Nash equilibrium and proves that a class of linear-quadratic differential games have equilibrium point when the duration of the game is sufficiently small.

Papavassilopoulos [374] proves existence and uniqueness of the solution for discrete-time linear-quadratic Gaussian Nash games with one-step delay observation sharing pattern. The solution is also linear in the information. Tu & Papavassilopoulos [501] consider discrete-time linear-quadratic Gaussian Nash games. They show that better information is beneficial to all players if the number of stages of the game, or the number of players, is larger than some bounds. For two-person zero-sum games, better information is beneficial to the player who has better maneuverability. Basar & Li [26] derive conditions for existence and uniqueness for stochastic linear-quadratic differential games. They also provide an algorithm for an iterative distributed computation of the solution.

When the classes are in equilibrium, no class can decrease its cost by altering its decision unilaterally. Next, we give the definition for a Nash equilibrium [27], for the join load sharing, routing and congestion control problem on the path flows.

Definition:

A vector $(\mathbf{\Phi}^*, \mathbf{\Psi}^*) \in (\mathbf{RC}, \mathbf{LS})$ *is called a* <u>*Nash equilibrium*</u> *for a C-class join load sharing, routing and congestion control problem if and only if*

$$J^1\left(\begin{array}{c}\mathbf{\Phi}^{1*}, ..., \mathbf{\Phi}^{c*}, ..., \mathbf{\Phi}^{C*} \\ \mathbf{\Psi}^{1*}, ..., \mathbf{\Psi}^{c*}, ..., \mathbf{\Psi}^{C*}\end{array}\right) \leq \inf_{\substack{\mathbf{\Phi}^1 \in \mathbf{RC}^1 \\ \mathbf{\Psi}^1 \in \mathbf{LS}^1}} J^1\left(\begin{array}{c}\mathbf{\Phi}^{1}, ..., \mathbf{\Phi}^{c*}, ..., \mathbf{\Phi}^{C*} \\ \mathbf{\Psi}^{1}, ..., \mathbf{\Psi}^{c*}, ..., \mathbf{\Psi}^{C*}\end{array}\right)$$

$$\cdots$$

$$J^c\left(\begin{array}{c}\mathbf{\Phi}^{1*}, ..., \mathbf{\Phi}^{c*}, ..., \mathbf{\Phi}^{C*} \\ \mathbf{\Psi}^{1*}, ..., \mathbf{\Psi}^{c*}, ..., \mathbf{\Psi}^{C*}\end{array}\right) \leq \inf_{\substack{\mathbf{\Phi}^c \in \mathbf{RC}^c \\ \mathbf{\Psi}^c \in \mathbf{LS}^c}} J^c\left(\begin{array}{c}\mathbf{\Phi}^{1*}, ..., \mathbf{\Phi}^{c}, ..., \mathbf{\Phi}^{C*} \\ \mathbf{\Psi}^{1*}, ..., \mathbf{\Psi}^{c}, ..., \mathbf{\Psi}^{C*}\end{array}\right)$$

$$\cdots$$

$$J^C\left(\begin{array}{c}\mathbf{\Phi}^{1*}, ..., \mathbf{\Phi}^{c*}, ..., \mathbf{\Phi}^{C*} \\ \mathbf{\Psi}^{1*}, ..., \mathbf{\Psi}^{c*}, ..., \mathbf{\Psi}^{C*}\end{array}\right) \leq \inf_{\substack{\mathbf{\Phi}^C \in \mathbf{RC}^C \\ \mathbf{\Psi}^C \in \mathbf{LS}^C}} J^C\left(\begin{array}{c}\mathbf{\Phi}^{1*}, ..., \mathbf{\Phi}^{c*}, ..., \mathbf{\Phi}^{C} \\ \mathbf{\Psi}^{1*}, ..., \mathbf{\Psi}^{c*}, ..., \mathbf{\Psi}^{C}\end{array}\right)$$

5.2.1 Optimal Control Formulation

In this section, we formulate the dynamic non-cooperative join load sharing, routing and congestion control problem as an Optimal Control Problem (OCP).

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 *and final time* t_f.

If for each class c, $H^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}(t))$ *is differentiable and convex in* $(\mathbf{X}, \mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{R}, \mathbf{RC}^c, \mathbf{LS}^c)$ $\forall\ t \in [t_0, t_f]$, *for each fixed value of*

$(\mathbf{\Phi}^1, \mathbf{\Psi}^1, ..., \mathbf{\Phi}^{c-1}, \mathbf{\Psi}^{c-1}, \mathbf{\Phi}^{c+1}, \mathbf{\Psi}^{c+1}, ..., \mathbf{\Phi}^C, \mathbf{\Psi}^C)$

$\in (\mathbf{RC}^1, \mathbf{LS}^1, ..., \mathbf{RC}^{c-1}, \mathbf{LS}^{c-1}, \mathbf{RC}^{c+1}, \mathbf{LS}^{c+1}, ..., \mathbf{RC}^C, \mathbf{LS}^C)$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a Nash equilibrium if and only if it solves the following* <u>*Optimal Control Problem*</u> $\forall\ t \in [t_0, t_f]$:

$\forall\ c$

$$\text{minimize} \quad \int_{t_0}^{t_f} g^c(t, \mathbf{X}(t), \mathbf{\Phi}^{1*}(t), \mathbf{\Psi}^{1*}(t), ..., \mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t), ..., \mathbf{\Phi}^{C*}(t), \mathbf{\Psi}^{C*}(t))dt$$

$$\text{with respect to} \quad (\mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t))$$

$$\text{such that} \quad \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$$

$$\mathbf{X}(t_0) = \mathbf{X}_0$$

$$(\mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t)) \in (\mathbf{RC}^c, \mathbf{LS}^c)$$

Proof: It follows from the definition of the Nash equilibrium. □

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 *and final time* t_f.

Let for each class c, $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *are continuously differentiable with respect to* $\mathbf{X} \in \mathbf{R}^n$, $\forall\ t \in [t_0, t_f]$.

If $(\hat{\mathbf{\Phi}}^*(t, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}_0)) = (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is an* <u>*open-loop*</u> <u>*Nash equilibrium*</u> *and* $\{\mathbf{X}^*(t),\ t \in [t_0, t_f]\}$ *is the corresponding state trajectory, then* $\exists\ \mathbf{P}^c(t) : [t_0, t_f] \rightarrow \mathbf{R}^n$, $\forall\ c$ *continuous and piecewise continuously differentiable vector functions, such that* $\forall\ t \in [t_0, t_f]$:

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\left[\frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{o[sd]}(t) = 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{\pi[sd]}(t) = 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c, c$$

$$\left[\frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)\right] * \psi^{c*}_{[sd]}(t) = 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}^c(t) = -\nabla_{\mathbf{X}} H^c(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}^c(t)) \quad \forall\ c$$

$$\mathbf{P}^c(t_f) = \mathbf{0} \qquad \forall\ c$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

Proof: The Lagrangian for each class c is

$$L^c = H^c + \sum_{[sd]\in \mathbf{SD}^c} Q^c_{[sd]} * \left[1 - \phi^c_{o[sd]} - \sum_{\pi[sd]\in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}\right] + \sum_{[s.]\in \mathbf{S}^c} Q^c_{[s.]} * \left[1 - \sum_{[.d]\in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]}\right]$$

with $\phi^c_{o[sd]},\ \phi^c_{\pi[sd]},\ \psi^c_{[sd]} \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$

Pontryagin's maximum principle necessary conditions are:

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\frac{\partial L^{c*}}{\partial \phi^c_{o[sd]}} * \phi^{c*}_{o[sd]}(t) = 0 \ \Rightarrow\ [\frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t)] * \phi^{c*}_{o[sd]}(t) = 0 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^{c*}}{\partial \phi^c_{\pi[sd]}} * \phi^{c*}_{\pi[sd]}(t) = 0 \ \Rightarrow\ [\frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t)] * \phi^{c*}_{\pi[sd]}(t) = 0$$

$\forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c, c$

$$\frac{\partial L^{c*}}{\partial \psi^c_{[sd]}} * \psi^{c*}_{[sd]}(t) = 0 \ \Rightarrow\ [\frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)] * \psi^{c*}_{[sd]}(t) = 0 \quad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\frac{\partial L^{c*}}{\partial \phi^c_{o[sd]}} \geq 0 \ \Rightarrow\ \frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t) \geq 0 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^{c*}}{\partial \phi^c_{\pi[sd]}} \geq 0 \ \Rightarrow\ \frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^{c*}}{\partial \psi^c_{[sd]}} \geq 0 \ \Rightarrow\ \frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}^c(t) = -\nabla_{\mathbf{X}} H^c(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}^c(t)) \qquad \forall\ c$$

$$\mathbf{P}^c(t_f) = \mathbf{0} \qquad \forall\ c$$

$$\frac{\partial L^{c*}}{\partial Q^c_{[sd]}} \geq 0 \ \Rightarrow \phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial L^{c*}}{\partial Q^c_{[s.]}} = 0 \ \Rightarrow \sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

□

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 *and final time* t_f.

Let for each class c, $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *are continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{RC}, \mathbf{LS})$, $\forall\ t \in [t_0, t_f]$.

If $(\hat{\mathbf{\Phi}}^*(t, \mathbf{X}, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}, \mathbf{X}_0)) = (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a closed-loop memoryless Nash equilibrium such that* $(\hat{\mathbf{\Phi}}^{c*}(t, \mathbf{X}, \mathbf{X}_0), \hat{\mathbf{\Phi}}^{c*}(t, \mathbf{X}, \mathbf{X}_0))$ *is continuously differentiable with respect to* $\mathbf{X} \in \mathbf{R}^n$, $\forall\ c,\ t \in [t_0, t_f]$ *and* $\{\mathbf{X}^*(t),\ t \in [t_0, t_f]\}$ *is the corresponding state trajectory, then* $\exists\ \mathbf{P}^c(t) : [t_0, t_f] \rightarrow \mathbf{R}^n$, $\forall\ c$, *continuous and piecewise continuously differentiable vector functions, such that* $\forall\ t \in [t_0, t_f]$:

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\left[\frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{o[sd]}(t) = 0 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t)\right] * \phi^{c*}_{\pi[sd]}(t) = 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c, c$$

$$\left[\frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t)\right] * \psi^{c*}_{[sd]}(t) = 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} - Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}^c(t) = -\nabla_{\mathbf{X}} H^c(t, \mathbf{X}^*, \hat{\mathbf{\Phi}}^*(t, \mathbf{X}^*, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}^*, \mathbf{X}_0), \mathbf{P}^c(t)) \quad \forall\ c$$

$$\mathbf{P}^c(t_f) = \mathbf{0} \qquad \forall\ c$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

Proof: The proof is similar to that for the open-loop solution.□

The above set of equations does not in general admit a single solution. In order to eliminate informational nonuniqueness in the derivation of Nash equilibrium under dynamic information, we constrain the Nash solution concept further (see next section).

5.2.2 Dynamic Programming Formulation

In this section, we formulate the dynamic non-cooperative joint load sharing, routing and congestion control problem as a Dynamic Programming Problem (DPP). Algorithms for solving DPP's may be found in books by Bellman [31], Howard [220], Kumar & Varaiya [274] Bertsekas [37], Ross [406] among others.

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 *and final time* t_f.

$(\Phi^*, \Psi^*) \in (\mathbf{RC}, \mathbf{LS})$ *is optimal if and only if the following conditions are satisfied:*

i) $\int_{t_0}^{t_f} g^c(t, \mathbf{X}^*(s), \hat{\Phi}^*(\mathbf{X}^*(s)), \hat{\Psi}^*(\mathbf{X}^*(s)) ds = constant \;\; \forall\, c$

ii) $\exists\, \mathbf{X}^*,\; \mathbf{P}^c,\;\; \forall\, c$ such that :

$$H^c(t, \mathbf{X}^*(t), \begin{array}{c} \Phi^{1*}(\mathbf{X}^*(t)), ..., \Phi^{c*}(\mathbf{X}^*(t)), ..., \Phi^{C*}(\mathbf{X}^*(t)) \\ \Psi^{1*}(\mathbf{X}^*(t)), ..., \Psi^{c*}(\mathbf{X}^*(t)), ..., \Psi^{C*}(\mathbf{X}^*(t)), \end{array} \mathbf{P}^c(t)) -$$

$$- H^c(t, \mathbf{X}(t), \begin{array}{c} \Phi^{1*}(\mathbf{X}(t)), ..., \Phi^{c}(\mathbf{X}(t)), ..., \Phi^{C*}(\mathbf{X}(t)), \\ \Psi^{1*}(\mathbf{X}(t)), ..., \Psi^{c}(\mathbf{X}(t)), ..., \Psi^{C*}(\mathbf{X}(t)), \end{array} \mathbf{P}^c(t)) +$$

$$+ \dot{\mathbf{P}}^c(t) * (\mathbf{X}^*(t) - \mathbf{X}) \;\; \leq 0 \;\; a.e. \;\; t \in [t_0, t_f],\; \forall\, \mathbf{X} \in \mathbf{R}^n,\; (\Phi^c, \Psi^c) \in (\mathbf{RC}^c, \mathbf{LS}^c),\; c$$

$$\mathbf{P}^c(t_f) * (\mathbf{X}^*(t_f) - \mathbf{X}) \leq 0 \quad \forall\, \mathbf{X} \in \mathbf{R}^n$$

Proof: By integration of ii) and using the state equation, we get the Nash equilibrium conditions.

Definition :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 *and final time* t_f.

Under the memoryless perfect state or closed-loop perfect state information structure, $(\hat{\Phi}, \hat{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$ *constitutes a* <u>*feedback Nash equilibrium*</u> *solution if and only if* $\exists\, V^c : [t_0, t_f] * \mathbf{R}^n \rightarrow R$ *satisfying the following relations for each class* c*:*

$$
\begin{aligned}
V^c(t,\mathbf{X}) &= \\
&= \int_t^{t_f} g^c(s,\mathbf{X}^*(s), \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{c*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{C*}(s,\mathbf{I}(s)) \\ \hat{\mathbf{\Psi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{c*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{C*}(s,\mathbf{I}(s)) \end{matrix})ds \leq \\
&\leq \int_t^{t_f} g^c(s,\mathbf{X}^*(s), \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{c}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{C*}(s,\mathbf{I}(s)) \\ \hat{\mathbf{\Psi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{c}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{C*}(s,\mathbf{I}(s)) \end{matrix})ds
\end{aligned}
$$

$$
\forall\ (\hat{\mathbf{\Phi}}^c(s,\mathbf{I}(s)),\hat{\mathbf{\Psi}}^c(s,\mathbf{I}(s))) \in (\mathbf{RC}^c,\mathbf{LS}^c),\ \mathbf{X}\in\mathbf{R}^n
$$

$$
\begin{aligned}
& \textit{such that}\ \ \forall\ s\in[t,t_f] \\
\dot{\mathbf{X}}^c(s) &= \mathbf{f}(s,\mathbf{X}^c(s), \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{c}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{C*}(s,\mathbf{I}(s)) \\ \hat{\mathbf{\Psi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{c}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{C*}(s,\mathbf{I}(s)) \end{matrix}) \\
\mathbf{X}^c(t) &= \mathbf{X} \\
\dot{\mathbf{X}}^*(s) &= \mathbf{f}(s,\mathbf{X}^*(s), \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{c*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Phi}}^{C*}(s,\mathbf{I}(s)) \\ \hat{\mathbf{\Psi}}^{1*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{c*}(s,\mathbf{I}(s)),...,\hat{\mathbf{\Psi}}^{C*}(s,\mathbf{I}(s)) \end{matrix}) \\
\mathbf{X}^*(s) &= \mathbf{X}
\end{aligned}
$$

where $\mathbf{I}(s) = \{\mathbf{X}(s),\mathbf{X}_0\}$ or $\mathbf{I}(s) = \{\mathbf{X}(\tau),\ \tau \leq s\}$.

The concept of feedback Nash equilibrium solution means that if $(\mathbf{\Phi}(s),\mathbf{\Psi}(s))$ is a feedback Nash equilibrium solution to the problem during $[t_0,t_f]$, is also a feedback Nash equilibrium solution to the problem during $[t,t_f]$, with the initial state taken as $\mathbf{X}(t)$. So, feedback Nash equilibrium strategies will depend only on the time variable and the current value of the state, but not on memory.

Proposition :

Every open-loop Nash equilibrium solution for the dynamic joint load sharing, routing and congestion control problem among cooperative classes is also closed-loop Nash equilibrium solution.

Proposition :

Under the memoryless (respectively, closed-loop) perfect state information structure, every feedback Nash equilibrium solution of the dynamic join load sharing, *routing and congestion control problem among competing classes is a closed-loop no memory (respectively, closed-loop) Nash equilibrium solution.*

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 and final time t_f.

Under the memory perfect state or closed loop perfect state information structure, $(\hat{\mathbf{\Phi}}, \hat{\mathbf{\Psi}}) \in (\mathbf{RC}, \mathbf{LS})$ *provides a* <u>*feedback Nash equilibrium solution*</u> *if* $\exists V^c : [t_0, t_f] * \mathbf{R}^n \rightarrow \mathbf{R},\ \forall\, c$ *satisfying the partial differential equations*

$$-\frac{\partial V^c(t,\mathbf{X})}{\partial t} =$$

$$= \min_{(\mathbf{\Phi}^c,\mathbf{\Psi}^c)\in(\mathbf{RC}^c,\mathbf{LS}^c)} \Big\{ \frac{\partial V^c(t,\mathbf{X})}{\partial \mathbf{X}} * \mathbf{f}(t,\mathbf{X}, \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(t,\mathbf{X}),...,\mathbf{\Phi}^c,...,\hat{\mathbf{\Phi}}^{C*}(t,\mathbf{X}) \\ \hat{\mathbf{\Psi}}^{1*}(t,\mathbf{X}),...,\mathbf{\Psi}^c,...,\hat{\mathbf{\Psi}}^{C*}(t,\mathbf{X}) \end{matrix})+$$

$$+\, g(t,\mathbf{X}, \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(t,\mathbf{X}),...,\mathbf{\Phi}^c,...,\hat{\mathbf{\Phi}}^{C*}(t,\mathbf{X}) \\ \hat{\mathbf{\Psi}}^{1*}(t,\mathbf{X}),...,\mathbf{\Psi}^c,...,\hat{\mathbf{\Psi}}^{C*}(t,\mathbf{X}) \end{matrix}) \Big\}$$

$$= \frac{\partial V^c(t,\mathbf{X})}{\partial \mathbf{X}} * \mathbf{f}(t,\mathbf{X}, \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(t,\mathbf{X}),...,\hat{\mathbf{\Phi}}^{c*}(t,\mathbf{X}),...,\hat{\mathbf{\Phi}}^{C*}(t,\mathbf{X}) \\ \hat{\mathbf{\Psi}}^{1*}(t,\mathbf{X}),...,\hat{\mathbf{\Psi}}^{c*}(t,\mathbf{X}),...,\hat{\mathbf{\Psi}}^{C*}(t,\mathbf{X}) \end{matrix})+$$

$$+\, g(t,\mathbf{X}, \begin{matrix} \hat{\mathbf{\Phi}}^{1*}(t,\mathbf{X}),...,\hat{\mathbf{\Phi}}^{c*}(t,\mathbf{X}),...,\hat{\mathbf{\Phi}}^{C*}(t,\mathbf{X}) \\ \hat{\mathbf{\Psi}}^{1*}(t,\mathbf{X}),...,\hat{\mathbf{\Psi}}^{c*}(t,\mathbf{X}),...,\hat{\mathbf{\Psi}}^{C*}(t,\mathbf{X}) \end{matrix})$$

5.2.3 Nonlinear Complementarity Problem Formulation

In this section, we formulate the dynamic non-cooperative load sharing, routing and congestion control problem as a Nonlinear Complementarity Problem (NCP).

Define the vector of class congestion control, routing and load sharing fractions as well as Lagrange multipliers:

$$\mathbf{Z}(t) = \left[...\ \phi^c_{o[sd]}(t)\ ...\ \phi^c_{\pi[sd]}\ ...\ Q^c_{[sd]}(t)\ ....\ \psi^c_{[sd]}(t)\ ...\ Q^c_{[s.]}(t)\ ...\right]^T$$

and the vector of class derivative of the Lagrangian with respect to the congestion control, routing and load sharing fractions as well as Lagrange multipliers:

$$\nabla L(\mathbf{Z}(t)) = \Bigg[\dots \left(\frac{\partial H^c}{\partial \phi^c_{o[sd]}} - Q^c_{o[sd]}(t) \right) \dots \left(\frac{\partial H^c}{\partial \phi^c_{\pi[sd]}} - Q^c_{\pi[sd]}(t) \right) \dots$$

$$\dots \left(1 - \phi^c_{o[sd]}(t) - \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}(t) \right) \dots$$

$$\dots \left(\frac{\partial H^c}{\partial \psi^c_{[sd]}} - Q^c_{[s.]}(t) \right) \dots \left(1 - \sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]}(t) \right) \dots \Bigg]$$

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 and final time t_f.

If for each class c,

g^c is differentiable and convex in $(\mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t)) \in (\mathbf{RC}^C, \mathbf{LS}^C)$, for each fixed value of $(\mathbf{\Phi}^1(t), \mathbf{\Psi}^1(t), \dots, \mathbf{\Phi}^{c-1}(t), \mathbf{\Psi}^{c-1}(t), \dots, \mathbf{\Psi}^{c+1}(t), \mathbf{\Phi}^{c+1}(t), \dots, \mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t))$

$\in (\mathbf{RC}^1, \mathbf{LS}^1, \dots, \mathbf{RC}^{c-1}, \mathbf{LS}^{c-1}, \dots, \mathbf{RC}^{c+1}, \mathbf{LS}^{c+1}, \dots, \mathbf{RC}^C, \mathbf{LS}^C)$

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ is a Nash equilibrium if and only if it solves the following Nonlinear Complementarity Problem $\forall\, t \in [t_0, t_f]$:

$\nabla L(\mathbf{Z}^*(t)) * \mathbf{Z}^*(t) = 0$

$\nabla L(\mathbf{Z}^*(t)) \geq 0$

$\mathbf{Z}^*(t) \geq 0$

$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$

$\mathbf{X}^*(t_0) = \mathbf{X}_0$

$\dot{\mathbf{P}}^c(t) = -\nabla_{\mathbf{X}} H^c(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}^c(t)) \quad \forall\, c$

$\mathbf{P}^c(t_f) = \mathbf{0} \quad \forall\, c$

Proof: After some algebraic manipulations, we find that the NCP: $\nabla L(\mathbf{Z}(t)) * \mathbf{Z}(t) = 0$; $\nabla L(\mathbf{Z}(t)) \geq 0$; $\mathbf{Z}(t) \geq 0$ with $\mathbf{Z}(t)$ and $\nabla L(\mathbf{Z}(t))$ as defined above, is equivalent to the Pontryagin's maximum principle necessary conditions. □

5.2.4 Variational Inequality Formulation

In this section, we formulate the dynamic non-cooperative load sharing, routing and congestion control problem as a Variational Inequality Problem (VIP).

Define the vector of class congestion control, routing and load sharing fractions:

$$(\mathbf{\Phi}(t), \mathbf{\Psi}(t)) = \left[\ldots \; \phi^c_{o[sd]}(t) \; \ldots \; \phi^c_{\pi[sd]}(t) \; \ldots \; \psi^c_{[sd]}(t) \; \ldots \right]^T$$

as well the vector of class derivatives of the cost function with respect to the congestion control, routing and load sharing fractions:

$$\nabla H(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t), \mathbf{P}(t)) \;=\; \left[\cdots \frac{\partial H^c}{\partial \phi^c_{o[sd]}} \cdots \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial H^c}{\partial \phi^c_{\pi[sd]}} \cdots \frac{\partial H^c}{\partial \psi^c_{[sd]}} \cdots \right]$$

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 *and final time* t_f.

Let for each class c, $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *be continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi}) \;\; \forall \, t \in [t_0, t_f]$. *If* H^c *is continuously differentiable and convex in* $(\mathbf{X}, \mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{R}^n, \mathbf{RC}^c, \mathbf{LS}^c)$, $\forall \, t \in [t_0, t_f]$, *for each fixed value of*

$(\mathbf{\Phi}^1(t), \mathbf{\Psi}^1(t), ..., \mathbf{\Phi}^{c-1}(t), \mathbf{\Psi}^{c-1}(t), \mathbf{\Phi}^{c+1}(t), \mathbf{\Psi}^{c+1}(t), ..., \mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t))$
$\in (\mathbf{RC}^1, \mathbf{LS}^1, ..., \mathbf{RC}^{c-1}, \mathbf{LS}^{c-1}, \mathbf{RC}^{c+1}, \mathbf{LS}^{c+1}, ..., \mathbf{RC}^C, \mathbf{LS}^C)$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a Nash equilibrium if and only if it solves the following* <u>*Variational Inequality Problem*</u> $\forall \, t \in [t_0, t_f]$:

$$\nabla H(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t)) * ((\mathbf{\Phi}, \mathbf{\Psi}) - (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))) \geq 0$$

$$\forall\ (\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$$

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\dot{\mathbf{P}}^c(t) = -\nabla_{\mathbf{X}} H^c(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}^c(t)) \quad \forall\ c$$

$$\mathbf{P}^c(t_f) = \mathbf{0} \quad \forall\ c$$

Proof: If $(\mathbf{\Phi}^{c*}(t), \mathbf{\Psi}^{c*}(t))$ is a local minimum for the following minimization problem

$$\begin{array}{ll} \textit{minimize} & \displaystyle\int_{t_0}^{t_f} g^c(t, \mathbf{X}(t), \begin{array}{c} \mathbf{\Phi}^{1*}(t), ..., \mathbf{\Phi}^c(t), ..., \mathbf{\Phi}^{C*}(t) \\ \mathbf{\Psi}^{1*}(t), ..., \mathbf{\Psi}^c(t), ..., \mathbf{\Psi}^{C*}(t) \end{array})dt \\ \textit{with respect to} & (\mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t)) \\ \textit{such that} & \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\ & \mathbf{X}(t_0) = \mathbf{X}_0 \\ & \\ & (\mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t)) \in (\mathbf{RC}^c, \mathbf{LS}^c) \end{array}$$

and g^c is a continuously differentiable convex function over the nonempty convex, closed and bounded set $(\mathbf{RC}^c, \mathbf{LS}^c)$, then $\forall \ddot{\in}[t_0, t_f]$:

$$\sum_{[sd]\in\mathbf{SD}^c} \left\{ \frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} * (\phi^c_{o[sd]} - \phi^{c*}_{o[sd]}(t)) + \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} * (\phi^c_{\pi[sd]} - \phi^{c*}_{\pi[sd]}(t)) + \right.$$

$$\left. + \frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} * (\psi^c_{[sd]} - \psi^{c*}_{[sd]}(t)) \right\} \geq 0 \quad \forall\ (\mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{RC}^c, \mathbf{LS}^c),\ c$$

Summing over all classes

$$\sum_c \sum_{[sd] \in \mathbf{SD}^c} \left\{ \frac{\partial H^{c*}}{\partial \phi^c_{o[sd]}} * (\phi^c_{o[sd]} - \phi^{c*}_{o[sd]}(t)) + \right.$$

$$+ \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial H^{c*}}{\partial \phi^c_{\pi[sd]}} * (\phi^c_{\pi[sd]} - \phi^{c*}_{\pi[sd]}(t)) +$$

$$\left. + \frac{\partial H^{c*}}{\partial \psi^c_{[sd]}} * (\psi^c_{[sd]} - \psi^{c*}_{[sd]}(t)) \right\} \geq 0 \qquad \forall \, (\mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{RC}^c, \mathbf{LS}^c)$$

□

Another equivalent VIP formulation is given in the following Theorem:

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with multiple competing classes, with fixed initial time t_0 and final time t_f.

Let for each class c, $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, be continuously differentiable with respect to $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi}) \;\; \forall \, t \in [t_0, t_f]$. If H^c is continuously differentiable and convex in $(\mathbf{X}, \mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{R}^n, \mathbf{RC}^c, \mathbf{LS}^c)$, $\;\forall \, t \in [t_0, t_f]$, for each fixed value of

$(\mathbf{\Phi}^1(t), \mathbf{\Psi}^1(t), ..., \mathbf{\Phi}^{c-1}(t), \mathbf{\Psi}^{c-1}(t), \mathbf{\Phi}^{c+1}(t), \mathbf{\Psi}^{c+1}(t), ..., \mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t))$
$\in (\mathbf{RC}^1, \mathbf{LS}^1, ..., \mathbf{RC}^{c-1}, \mathbf{LS}^{c-1}, \mathbf{RC}^{c+1}, \mathbf{LS}^{c+1}, ..., \mathbf{RC}^C, \mathbf{LS}^C)$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ is a Nash equilibrium if and only if it solves the following Variational Inequality Problem $\forall \, t \in [t_0, t_f]$:

$$\nabla L(\mathbf{Z}(t)^*) * (\mathbf{Z} - \mathbf{Z}(t)^*) \geq 0 \qquad \forall \, \mathbf{Z} \geq 0$$
$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$
$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$
$$\dot{\mathbf{P}}^c(t) = -\nabla_{\mathbf{X}} H^c(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}^c(t)) \quad \forall \, c$$
$$\mathbf{P}^c(t_f) = \mathbf{0} \quad \forall \, c$$

Proof: The NCP: $f(x^*) * x^* = 0 \;\; f(x^*) \geq 0 \;\; x^* \geq 0$
and the VIP: find x^* such that $f(x^*) * (x - x^*) \geq 0 \;\; \forall x \geq 0$
are equivalent.□

5.2.5 Maximum Principle for Separable Cost Functions

In this section, we derive the first order necessary conditions for a Nash equilibrium on the path flows, when the cost function of each resource depends only on the flow on this resource.

According to the Nash equilibrium definition, each class c minimizes its cost function g^c given the optimum decisions of all other classes.

$$\text{minimize} \quad \int_{t_0}^{t_f} g^c(t, \mathbf{X}(t), \begin{matrix} \mathbf{\Phi}^{1*}(t), ..., \mathbf{\Phi}^{c}(t), ..., \mathbf{\Phi}^{C*}(t) \\ \mathbf{\Psi}^{1*}(t), ..., \mathbf{\Psi}^{c}(t), ..., \mathbf{\Psi}^{C*}(t) \end{matrix})dt =$$

$$= \sum_{ij} \int_{t_0}^{t_f} g_{ij}^c(t, \mathbf{X}_{ij}(t), \lambda_{ij}^{1*}(t), ..., \lambda_{ij}^{c}(t), ..., \lambda_{ij}^{C*}(t))dt \ +$$

$$+ \ \sum_{i} \int_{t_0}^{t_f} g_{i}^c(t, \mathbf{X}_{i}(t), \lambda_{i}^{1*}(t), ..., \lambda_{i}^{c}(t), ..., \lambda_{i}^{C*}(t))dt \ +$$

$$+ \ \sum_{[sd]} \int_{t_0}^{t_f} g_{[sd]}^c(t, \mathbf{X}_{o[sd]}(t), \lambda_{o[sd]}^{1*}(t), ..., \lambda_{o[sd]}^{c}(t), ..., \lambda_{o[sd]}^{C*}(t))dt \ +$$

$$+ \ \sum_{[.d]} \int_{t_0}^{t_f} g_{[.d]}^c(t, \mathbf{X}_{[.d]}(t), \lambda_{[.d]}^{1*}(t), ..., \lambda_{[.d]}^{c}(t), ..., \lambda_{[.d]}^{C*}(t))dt$$

$$\text{with respect to} \quad (\mathbf{\Phi}^c(t), \mathbf{\Psi}^c(t))$$

such that

$$\begin{aligned}
\dot{\mathbf{X}}^k_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\phi^c_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c$$

$$\phi^c_{o[sd]}(t),\ \phi^c_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c$$

$$\psi^c_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c$$

Pontryagin's maximum principle necessary conditions are:

$$
\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^{k}_{ij[sd]}(t, \mathbf{X}^{*}_{ij}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^{k}_{i[sd]}(t, \mathbf{X}^{*}_{i}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^{k}_{o[sd]}(t, \mathbf{X}^{*}_{o[sd]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^{k}_{[.d][sd]}(t, \mathbf{X}^{*}_{[.d]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^{k}_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^{k}_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^{k}_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^{k},\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^{k}_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^{k},\ k
\end{aligned}
$$

$$\left[\frac{\partial g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{o[sd]}} + \right.$$

$$+ \sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\left. - Q^c_{[sd]}(t) \right] * \phi^{c*}_{o[sd]}(t) = 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\sum_{ij} \frac{\partial g^c_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} + \sum_i \frac{\partial g^c_i(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} + \right.$$

$$+ \sum_k \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_k \sum_{[s'd']} \sum_i \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\left. - Q^c_{[sd]}(t) \right] * \phi^{c*}_{\pi[sd]}(t) = 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$
\left[\sum_{ij}\frac{\partial g^c_{ij}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))}{\partial\psi^c_{[sd]}}+\sum_{i}\frac{\partial g^c_{i}(t,\mathbf{X}^*_{i}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))}{\partial\psi^c_{[sd]}}+\right.
$$

$$
+\frac{\partial g^c_{[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))}{\partial\psi^c_{[sd]}}+\frac{\partial g^c_{[.d]}(t,\mathbf{X}^*_{[.d]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))}{\partial\psi^c_{[sd]}}+
$$

$$
+\sum_{k}\sum_{[s'd']}\sum_{ij}\mathbf{P}^{k,c}_{ij[s'd']}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{ij[s'd']}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+
$$

$$
+\sum_{k}\sum_{[s'd']}\sum_{i}\mathbf{P}^{k,c}_{i[s'd']}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{i[s'd']}(t,\mathbf{X}^*_{i}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+
$$

$$
+\sum_{k}\mathbf{P}^{k,c}_{o[sd]}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{o[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+
$$

$$
+\sum_{k}\sum_{[s'.]}\mathbf{P}^{k,c}_{[.d][s'd]}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{[.d][s'd]}(t,\mathbf{X}^*_{[.d]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))-
$$

$$
\left.-Q^c_{[s.]}(t)\right]*\psi^{c*}_{[sd]}(t)=0 \qquad \forall\ [.d]\in\mathbf{D}^c_{[s.]},\ [s.]\in\mathbf{S}^c,\ c
$$

$$\frac{\partial g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{o[sd]}} +$$

$$\sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \frac{\partial g^c_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} + \sum_i \frac{\partial g^c_i(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} +$$

$$+ \sum_k \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_k \sum_{[s'd']} \sum_i \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \frac{\partial g_{ij}^c(t, \mathbf{X}_{ij}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi_{[sd]}^c} + \sum_{i} \frac{\partial g_{i}^c(t, \mathbf{X}_{i}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi_{[sd]}^c} +$$

$$+ \frac{\partial g_{[sd]}^c(t, \mathbf{X}_{o[sd]}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi_{[sd]}^c} + \frac{\partial g_{[.d]}^c(t, \mathbf{X}_{[.d]}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi_{[sd]}^c} +$$

$$+ \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}_{ij[s'd']}^{k,c}(t) * \nabla_{\psi_{[sd]}^c} \mathbf{f}_{ij[s'd']}^k(t, \mathbf{X}_{ij}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}_{i[s'd']}^{k,c}(t) * \nabla_{\psi_{[sd]}^c} \mathbf{f}_{i[s'd']}^k(t, \mathbf{X}_{i}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \mathbf{P}_{o[sd]}^{k,c}(t) * \nabla_{\psi_{[sd]}^c} \mathbf{f}_{o[sd]}^k(t, \mathbf{X}_{o[sd]}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \sum_{[s'.]} \mathbf{P}_{[.d][s'd]}^{k,c}(t) * \nabla_{\psi_{[sd]}^c} \mathbf{f}_{[.d][s'd]}^k(t, \mathbf{X}_{[.d]}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-Q_{[s.]}^c(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}_{[s.]}^c,\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g^c_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g^c_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{i[s'd']}(t) * \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \mathbf{P}^{c,n}_{o[sd]}(t) * \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g^c_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

The partial derivatives of the cost function $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$ with respect to the path fractions $\phi^c_{\pi[sd]}$ can be written with respect to the link flows λ^c_{ij} and node flows λ^c_i:

$$\frac{\partial g^c_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^c_{\pi[sd]}} = \frac{\partial g^c_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda^c_{ij}} * \frac{\partial \lambda^c_{ij}}{\partial \phi^c_{\pi[sd]}} =$$

$$= \frac{\partial g^c_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda^c_{ij}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}(t)) * 1_{ij \in \pi[sd]}(t)$$

$$\frac{\partial g^c_i(t, \mathbf{X}_i, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^c_{\pi[sd]}} = \frac{\partial g^c_i(t, \mathbf{X}_i, \mathbf{\Lambda}_i)}{\partial \lambda^c_i} * \frac{\partial \lambda^c_i}{\partial \phi^c_{\pi[sd]}} =$$

$$= \frac{\partial g^c_i(t, \mathbf{X}_i, \mathbf{\Lambda}_i)}{\partial \lambda^c_i} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}(t)) * 1_{i \in \pi[sd]}(t)$$

$$\frac{\partial g^c_{[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^c_{o[sd]}} = \frac{\partial g^c_{[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda^c_{o[sd]}} * \frac{\partial \lambda^c_{o[sd]}}{\partial \phi^c_{o[sd]}} =$$

$$= \frac{\partial g^c_{[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}(t))$$

$$\frac{\partial g_{ij}^c(t, \mathbf{X}_{ij}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^c} = \frac{\partial g_{ij}^c(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda_{ij}^c} * \frac{\partial \lambda_{ij}^c}{\partial \psi_{[sd]}^c} =$$

$$= \sum_{\pi[sd] \in \mathbf{\Pi}_{[sd]}^c} \frac{\partial g_{ij}^c(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda_{ij}^c} * \gamma_{[s.]}^c(t) * \phi_{\pi[sd]}^c(t) * 1_{ij \in \pi[sd]}(t)$$

$$\frac{\partial g_{i}^c(t, \mathbf{X}_{i}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^c} = \frac{\partial g_{i}^c(t, \mathbf{X}_{i}, \mathbf{\Lambda}_{i})}{\partial \lambda_{i}^c} * \frac{\partial \lambda_{i}^c}{\partial \psi_{[sd]}^c} =$$

$$= \sum_{\pi[sd] \in \mathbf{\Pi}_{[sd]}^c} \frac{\partial g_{i}^c(t, \mathbf{X}_{i}, \mathbf{\Lambda}_{i})}{\partial \lambda_{i}^c} * \gamma_{[s.]}^c(t) * \phi_{\pi[sd]}^c(t) * 1_{i \in \pi[sd]}(t)$$

$$\frac{\partial g_{[sd]}^c(t, \mathbf{X}_{o[sd]}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^c} = \frac{\partial g_{[sd]}^c(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda_{o[sd]}^c} * \frac{\partial \lambda_{o[sd]}^c}{\partial \psi_{[sd]}^c} =$$

$$= \frac{\partial g_{[sd]}^c(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda_{o[sd]}^c} * \gamma_{[s.]}^c(t) * \phi_{o[sd]}^c(t)$$

$$\frac{\partial g_{[.d]}^c(t, \mathbf{X}_{[.d]}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^c} = \frac{\partial g_{[.d]}^c(t, \mathbf{X}_{[.d]}, \mathbf{\Lambda}_{[.d]})}{\partial \lambda_{[.d]}^c} * \frac{\partial \lambda_{[.d]}^c}{\partial \psi_{[sd]}^c} =$$

$$= \frac{\partial g_{[.d]}^c(t, \mathbf{X}_{[.d]}, \mathbf{\Lambda}_{[.d]})}{\partial \lambda_{[.d]}^c} * \gamma_{[s.]}^c(t)$$

Then Pontryagin's maximum principle becomes:

$$\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) = \mathbf{f}^{k}_{ij[sd]}(t, \mathbf{X}^{*}_{ij}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) \quad \forall\, ij,\ [sd] \in \mathbf{SD}^{k},\ k$$

$$\dot{\mathbf{X}}^{k*}_{i[sd]}(t) = \mathbf{f}^{k}_{i[sd]}(t, \mathbf{X}^{*}_{i}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) \quad \forall \subset,\ [sd] \in \mathbf{SD}^{k},\ k$$

$$\dot{\mathbf{X}}^{k*}_{o[sd]}(t) = \mathbf{f}^{k}_{o[sd]}(t, \mathbf{X}^{*}_{o[sd]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) \quad \forall\, [sd] \in \mathbf{SD}^{k},\ k$$

$$\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) = \mathbf{f}^{k}_{[.d][sd]}(t, \mathbf{X}^{*}_{[.d]}(t), \mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) \quad \forall\, [sd] \in \mathbf{SD}^{k},\ k$$

$$\mathbf{X}^{k*}_{ij[sd]}(t_0) = \mathbf{X}^{k}_{ij[sd],0} \quad \forall\, ij,\ [sd] \in \mathbf{SD}^{k},\ k$$

$$\mathbf{X}^{k*}_{i[sd]}(t_0) = \mathbf{X}^{k}_{i[sd],0} \quad \forall\, i,\ [sd] \in \mathbf{SD}^{k},\ k$$

$$\mathbf{X}^{k*}_{o[sd]}(t_0) = \mathbf{X}^{k}_{o[sd],0} \quad \forall\, [sd] \in \mathbf{SD}^{k},\ k$$

$$\mathbf{X}^{k*}_{[.d][sd]}(t_0) = \mathbf{X}^{k}_{[.d][sd],0} \quad \forall\, [sd] \in \mathbf{SD}^{k},\ k$$

$$\left[\frac{\partial g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Lambda}^*_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) + \right.$$

$$+ \sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\left. - Q^c_{[sd]}(t) \right] * \phi^{c*}_{o[sd]}(t) = 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\sum_{ij} \frac{\partial g^c_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Lambda}^*_{ij}(t))}{\partial \lambda^c_{ij}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{ij \in \pi[sd]}(t) + \right.$$

$$+ \sum_i \frac{\partial g^c_i(t, \mathbf{X}^*_i(t), \mathbf{\Lambda}^*_i(t))}{\partial \lambda^c_i} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{i \in \pi[sd]}(t) +$$

$$+ \sum_k \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_k \sum_{[s'd']} \sum_i \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$\left. -Q^c_{[sd]}(t) \right] * \phi^{c*}_{\pi[sd]}(t) = 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\left[\sum_{ij}\sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}}\frac{\partial g^c_{ij}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Lambda}^*_{ij}(t))}{\partial\lambda^c_{ij}}*\gamma^c_{[s.]}(t)*\phi^{c*}_{\pi[sd]}(t))*1_{ij\in\pi[sd]}(t)+\right.$$

$$+\sum_{i}\sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}}\frac{\partial g^c_{i}(t,\mathbf{X}^*_{i}(t),\mathbf{\Lambda}^*_{i}(t))}{\partial\lambda^c_{i}}*\gamma^c_{[s.]}(t)*\phi^{c*}_{\pi[sd]}(t))*1_{i\in\pi[sd]}(t)+$$

$$+\frac{\partial g^c_{[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Lambda}^*_{o[sd]}(t))}{\partial\lambda^c_{o[sd]}}*\gamma^c_{[s.]}(t)*\phi^{c*}_{o[sd]}(t))+$$

$$+\frac{\partial g^c_{[.d]}(t,\mathbf{X}^*_{[.d]}(t),\mathbf{\Lambda}^*_{[.d]}(t))}{\partial\lambda^c_{[.d]}}*\gamma^c_{[s.]}(t)+$$

$$+\sum_k\sum_{[s'd']}\sum_{ij}\mathbf{P}^{k,c}_{ij[s'd']}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{ij[s'd']}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+$$

$$+\sum_k\sum_{[s'd']}\sum_{i}\mathbf{P}^{k,c}_{i[s'd']}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{i[s'd']}(t,\mathbf{X}^*_{i}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+$$

$$+\sum_k\mathbf{P}^{k,c}_{o[sd]}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{o[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))+$$

$$+\sum_k\sum_{[s'.]}\mathbf{P}^{k,c}_{[.d][s'd]}(t)*\nabla_{\psi^c_{[sd]}}\mathbf{f}^k_{[.d][s'd]}(t,\mathbf{X}^*_{[.d]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t))-$$

$$\left.-Q^c_{[s.]}(t)\right]*\psi^{c*}_{[sd]}(t)=0 \qquad \forall\ [.d]\in\mathbf{D}^c_{[s.]},\ [s.]\in\mathbf{S}^c,\ c$$

$$\frac{\partial g^c_{[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Lambda}^*_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) +$$

$$+ \sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t,\mathbf{X}^*_{o[sd]}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t)) -$$

$$-Q^c_{[sd]}(t) \geq 0 \qquad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \frac{\partial g^c_{ij}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Lambda}^*_{ij}(t))}{\partial \lambda^c_{ij}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{ij\in\pi[sd]}(t) +$$

$$+ \sum_i \frac{\partial g^c_i(t,\mathbf{X}^*_i(t),\mathbf{\Lambda}^*_i(t))}{\partial \lambda^c_i} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^{c*}_{[sd]}(t)) * 1_{i\in\pi[sd]}(t) +$$

$$+ \sum_k \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t,\mathbf{X}^*_{ij}(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t)) +$$

$$+ \sum_k \sum_{[s'd']} \sum_i \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t,\mathbf{X}^*_i(t),\mathbf{\Phi}^*(t),\mathbf{\Psi}^*(t)) -$$

$$-Q^c_{[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{ij} \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g^c_{ij}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Lambda}^*_{ij}(t))}{\partial \lambda^c_{ij}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{\pi[sd]}(t)) * 1_{ij \in \pi[sd]}(t) +$$

$$+ \sum_{i} \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g^c_{i}(t, \mathbf{X}^*_{i}(t), \mathbf{\Lambda}^*_{i}(t))}{\partial \lambda^c_{i}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{\pi[sd]}(t)) * 1_{i \in \pi[sd]}(t) +$$

$$+ \frac{\partial g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Lambda}^*_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * \gamma^c_{[s.]}(t) * \phi^{c*}_{o[sd]}(t)) +$$

$$+ \frac{\partial g^c_{[.d]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Lambda}^*_{[.d]}(t))}{\partial \lambda^c_{[.d]}} * \gamma^c_{[s.]}(t) +$$

$$+ \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{i[s'd']}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \mathbf{P}^{k,c}_{o[sd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) +$$

$$+ \sum_{k} \sum_{[s'.]} \mathbf{P}^{k,c}_{[.d][s'd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$- Q^c_{[s.]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g^c_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g^c_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \mathbf{P}^{c,n}_{i[s'd']}(t) * \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \mathbf{P}^{c,n}_{o[sd]}(t) * \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g^c_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$\sum_{[.d]\in\mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^c,\ c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c,\ c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c$$

Next, for each class c, we define the length for the rejected flow $[sd]$, the length for the path $\pi[sd]$ and the length for the source-destination pair $[sd]$:

$$l^{c,Nash}_{o[sd]}(t) = \frac{\partial g^c_{[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Lambda}_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}(t)) +$$

$$+ \sum_k \mathbf{P}^{k,c}_{o[sd]} * \nabla_{\phi^c_{o[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$$

$$\forall\ [sd] \in \mathbf{SD}^c,\ c$$

$$
\begin{aligned}
l^{c,Nash}_{\pi[sd]}(t) = \; & \sum_{ij} \frac{\partial g^c_{ij}(t, \mathbf{X}_{ij}(t), \mathbf{\Lambda}_{ij}(t))}{\partial \lambda^c_{ij}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}(t)) * 1_{ij \in \pi[sd]}(t) + \\
& + \sum_{i} \frac{\partial g^c_{i}(t, \mathbf{X}_{i}(t), \mathbf{\Lambda}_{i}(t))}{\partial \lambda^c_{i}} * (\gamma^c_{[sd]}(t) + \gamma^c_{[s.]}(t) * \psi^c_{[sd]}(t)) * 1_{i \in \pi[sd]}(t) + \\
& + \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) + \\
& + \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\phi^c_{\pi[sd]}} \mathbf{f}^c_{i[s'd']}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))
\end{aligned}
$$

$$
\forall \; \pi[sd] \in \mathbf{\Pi}^c_{[sd]}, \; [sd] \in \mathbf{SD}^c, \; c
$$

$$
\begin{aligned}
l^{c,Nash}_{[sd]}(t) = & \sum_{ij} \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g^c_{ij}(t, \mathbf{X}_{ij}(t), \mathbf{\Lambda}_{ij}(t))}{\partial \lambda^c_{ij}} * \gamma^c_{[s.]}(t) * \phi^c_{\pi[sd]}(t)) * 1_{ij \in \pi[sd]}(t) + \\
& + \sum_{i} \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g^c_{i}(t, \mathbf{X}_{i}(t), \mathbf{\Lambda}_{i}(t))}{\partial \lambda^c_{i}} * \gamma^c_{[s.]}(t) * \phi^c_{\pi[sd]}(t)) * 1_{i \in \pi[sd]}(t) + \\
& + \frac{\partial g^c_{[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Lambda}_{o[sd]}(t))}{\partial \lambda^c_{o[sd]}} * \gamma^c_{[s.]}(t) * \phi^c_{o[sd]}(t)) + \\
& + \frac{\partial g^c_{[.d]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Lambda}_{[.d]}(t))}{\partial \lambda^c_{[.d]}} * \gamma^c_{[s.]}(t) + \\
& + \sum_{k} \sum_{[s'd']} \sum_{ij} \mathbf{P}^{k,c}_{ij[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{ij[s'd']}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) + \\
& + \sum_{k} \sum_{[s'd']} \sum_{i} \mathbf{P}^{k,c}_{i[s'd']}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{i[s'd']}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) + \\
& + \sum_{k} \mathbf{P}^{k,c}_{o[sd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) + \\
& + \sum_{k} \sum_{[s'.]} \mathbf{P}^{k,c}_{[.d][s'd]}(t) * \nabla_{\psi^c_{[sd]}} \mathbf{f}^k_{[.d][s'd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) - \\
& \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c,\ c
\end{aligned}
$$

External arriving flow at a source is assigned to the destination that has the minimum length from the source. However, this flow may be rejected if the length of rejecting it is less than the lengths of the paths to its destination. If it is accepted, then it is routed to its destination via the minimum length path.

In the next section, we will derive the same conditions by an alternative way, and we shall state the above ideas more formally.

5.2.6 V.I. for Separable Cost Functions

Equivalently, the Nash equilibrium definition, each class c minimizes its cost function g^c given the optimum decisions of all other classes. We first solve the routing and congestion control problem assuming that all other classes act optimally for themselves. So, class c first solves the routing and congestion control problems

$$\text{minimize} \qquad \int_{t_0}^{t_f} g^c(t, \mathbf{X}, \begin{array}{c} \mathbf{\Phi}^{1*}(t), ..., \mathbf{\Phi}^{c}(t), ..., \mathbf{\Phi}^{C*}(t) \\ \mathbf{\Psi}^{1*}(t), ..., \mathbf{\Psi}^{c}(t), ..., \mathbf{\Psi}^{C*}(t) \end{array})dt =$$

$$= \sum_{ij} \int_{t_0}^{t_f} g_{ij}^c(t, \mathbf{X}_{ij}(t), \lambda_{ij}^{1*}(t), ..., \lambda_{ij}^{c}(t), ..., \lambda_{ij}^{C*}(t))dt \ +$$

$$+ \ \sum_{i} \int_{t_0}^{t_f} g_{i}^c(t, \mathbf{X}_{i}(t), \lambda_{i}^{1*}(t), ..., \lambda_{i}^{c}(t), ..., \lambda_{i}^{C*}(t))dt \ +$$

$$+ \ \sum_{[sd]} \int_{t_0}^{t_f} g_{[sd]}^c(t, \mathbf{X}_{o[sd]}(t), \lambda_{o[sd]}^{1*}(t), ..., \lambda_{o[sd]}^{c}(t), ..., \lambda_{o[sd]}^{C*}(t))dt \ +$$

$$+ \ \sum_{[.d]} \int_{t_0}^{t_f} g_{[.d]}^c(t, \mathbf{X}_{[.d]}(t), \lambda_{[.d]}^{1*}(t), ..., \lambda_{[.d]}^{c}(t), ..., \lambda_{[.d]}^{C*}(t))dt$$

with respect to $\mathbf{\Phi}^c(t)$

such that

$$
\begin{aligned}
\dot{\mathbf{X}}^k_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}
$$

$$
\phi^c_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^c_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c
$$

$$
\phi^c_{o[sd]}(t),\ \phi^c_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c
$$

The necessary optimality conditions for class c are

$$\sum_{[sd]\in \mathbf{SD}^c} \left\{ \frac{\partial g^c(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{o[sd]}} * (\phi^c_{o[sd]} - \phi^{c*}_{o[sd]}(t)) \; + \right.$$

$$\left. + \sum_{\pi[sd]\in \mathbf{\Pi}^c_{[sd]}} \frac{\partial g^c(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} * (\phi^c_{\pi[sd]} - \phi^{c*}_{\pi[sd]}(t)) \right\} \geq 0 \qquad \forall\ \mathbf{\Phi}^c \in \mathbf{RC}^c$$

such that

$$\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) = \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{i[sd]}(t) = \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_i(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{o[sd]}(t) = \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) = \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{ij[sd]}(t_0) = \mathbf{X}^k_{ij[sd],0} \qquad \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{i[sd]}(t_0) = \mathbf{X}^k_{i[sd],0} \qquad \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{o[sd]}(t_0) = \mathbf{X}^k_{o[sd],0} \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{[.d][sd]}(t_0) = \mathbf{X}^k_{[.d][sd],0} \qquad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g^c_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g^c_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g^c_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^c$$

$$\phi^{c*}_{o[sd]}(t),\ \phi^{c*}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]},\ [sd] \in \mathbf{SD}^c$$

We can decompose these conditions for each source-destination pair $[sd] \in \mathbf{SD}^c$

$$\frac{\partial g^c(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{o[sd]}} * (\phi^c_{o[sd]} - \phi^{c*}_{o[sd]}(t)) +$$

$$+ \sum_{\pi[sd]\in\mathbf{\Pi}^c_{[sd]}} \frac{\partial g^c(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \phi^c_{\pi[sd]}} * (\phi^c_{\pi[sd]} - \phi^{c*}_{\pi[sd]}(t)) \geq 0 \qquad \forall\ \mathbf{\Phi}^c \in \mathbf{RC}^c$$

such that

$$\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) = \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \quad \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{i[sd]}(t) = \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \quad \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{o[sd]}(t) = \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \quad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) = \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \quad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{ij[sd]}(t_0) = \mathbf{X}^k_{ij[sd],0} \quad \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{i[sd]}(t_0) = \mathbf{X}^k_{i[sd],0} \quad \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{o[sd]}(t_0) = \mathbf{X}^k_{o[sd],0} \quad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\mathbf{X}^{k*}_{[.d][sd]}(t_0) = \mathbf{X}^k_{[.d][sd],0} \quad \forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g^c_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ ij,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g^c_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g^c_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^c_{[sd]}} \phi^{c*}_{\pi[sd]} = 1$$

$$\phi^c_{o[sd]}(t),\ \phi^c_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^c_{[sd]}$$

Theorem : Routing

There must be flow only on minimum length paths:

$$\phi^{c*}_{\pi[sd]}(t) > 0 \quad only \quad if \qquad l^{c,Nash*}_{\pi[sd]}(t) = \min\{l^{c,Nash*}_{o[sd]}(t), \min_{p[sd]}\{l^{c,Nash*}_{p[sd]}(t)\}\}$$

$$\phi^{c*}_{\pi[sd]}(t) = 0 \qquad\qquad o.w.$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^{c}_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1$$

$$\forall\ \pi[sd] \in \Pi^{c}_{[sd]}, \qquad [sd] \in \mathbf{SD}^{c}, \quad c$$

and satisfies the partial differential vectors for the state and the costate variables.

Theorem : Congestion Control

Flow is not admitted into the network only if its rejection length is less than the minimum length path to its destination:

$$\phi^{c*}_{o[sd]}(t) > 0 \quad only \quad if \qquad l^{c,Nash}_{o[sd]}(t) = \min\{l^{c,Nash*}_{o[sd]}(t), \min_{p[sd]}\{l^{c,Nash*}_{p[sd]}(t)\}\}$$

$$\phi^{c*}_{\pi[sd]} = 0 \qquad\qquad o.w.$$

$$\phi^{c*}_{o[sd]}(t) + \sum_{\pi[sd]\in\mathbf{\Pi}^{c}_{[sd]}} \phi^{c*}_{\pi[sd]}(t) = 1 \ \ \forall\ [sd] \in \mathbf{SD}^{c}, \quad c$$

and satisfies the partial differential vectors for the state and the costate variables.

Having found the optimum routing and congestion control decisions, we proceed to solve the load sharing problem for class c assuming also that all other classes act at their optimum decisions. So, the load sharing problem for class c is

$$
\textit{minimize} \qquad \int_{t_0}^{t_f} g^c\left(t, \mathbf{X}(t), \begin{array}{c} \mathbf{\Phi}^{1*}(t), ..., \mathbf{\Phi}^{c*}(t), ..., \mathbf{\Phi}^{C*}(t) \\ \mathbf{\Psi}^{1*}(t), ..., \mathbf{\Psi}^{c}(t), ..., \mathbf{\Psi}^{C*}(t) \end{array}\right) dt =
$$

$$
= \sum_{ij} \int_{t_0}^{t_f} g_{ij}^c(t, \mathbf{X}_{ij}(t), \lambda_{ij}^{1*}(t), ..., \lambda_{ij}^{c}(t), ..., \lambda_{ij}^{C*}(t))dt \ +
$$

$$
+ \ \sum_{i} \int_{t_0}^{t_f} g_{i}^c(t, \mathbf{X}_{i}(t), \lambda_{i}^{1*}(t), ..., \lambda_{i}^{c}(t), ..., \lambda_{i}^{C*}(t))dt \ +
$$

$$
+ \ \sum_{[sd]} \int_{t_0}^{t_f} g_{[sd]}^c(t, \mathbf{X}_{o[sd]}(t), \lambda_{o[sd]}^{1*}(t), ..., \lambda_{o[sd]}^{c}(t), ..., \lambda_{o[sd]}^{C*}(t))dt \ +
$$

$$
+ \ \sum_{[.d]} \int_{t_0}^{t_f} g_{[.d]}^c(t, \mathbf{X}_{[.d]}(t), \lambda_{[.d]}^{1*}(t), ..., \lambda_{[.d]}^{c}(t), ..., \lambda_{[.d]}^{C*}(t))dt
$$

with respect to $\mathbf{\Psi}^c(t)$

such that

$$\begin{aligned}
\dot{\mathbf{X}}^k_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}_{ij}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}_{i}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}_{o[sd]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^k_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}_{[.d]}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^k_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\sum_{[.d]\in \mathbf{D}^c_{[s.]}} \psi^c_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c$$

$$\psi^c_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c$$

The necessary optimality conditions for class c are:

$$\frac{\partial g^c(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} * (\psi^c_{[sd]} - \psi^{c*}_{[sd]}(t)) \quad \geq 0 \qquad \forall\ \mathbf{\Psi}^c \in \mathbf{LS}^c$$

such

$$\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g^c_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g^c_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g^c_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]},\ [s.] \in \mathbf{S}^c$$

We can decompose these conditions for each source node $[s.] \in \mathbf{S}^c$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \frac{\partial g^c(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))}{\partial \psi^c_{[sd]}} * (\psi^c_{[sd]} - \psi^{c*}_{[sd]}(t)) \quad \geq 0 \quad \forall\ \mathbf{\Psi}^c \in \mathbf{LS}^c$$

such that

$$\begin{aligned}
\dot{\mathbf{X}}^{k*}_{ij[sd]}(t) &= \mathbf{f}^k_{ij[sd]}(t, \mathbf{X}^*_{ij}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{i[sd]}(t) &= \mathbf{f}^k_{i[sd]}(t, \mathbf{X}^*_{i}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{o[sd]}(t) &= \mathbf{f}^k_{o[sd]}(t, \mathbf{X}^*_{o[sd]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\dot{\mathbf{X}}^{k*}_{[.d][sd]}(t) &= \mathbf{f}^k_{[.d][sd]}(t, \mathbf{X}^*_{[.d]}(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{ij[sd]}(t_0) &= \mathbf{X}^k_{ij[sd],0} && \forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{i[sd]}(t_0) &= \mathbf{X}^k_{i[sd],0} && \forall\ i,\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{o[sd]}(t_0) &= \mathbf{X}^k_{o[sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k \\
\mathbf{X}^{k*}_{[.d][sd]}(t_0) &= \mathbf{X}^k_{[.d][sd],0} && \forall\ [sd] \in \mathbf{SD}^k,\ k
\end{aligned}$$

$$\dot{\mathbf{P}}^{c,k}_{ij[sd]}(t) = -\nabla_{\mathbf{X}^k_{ij[sd]}} g^c_{ij}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{ij[sd]}} \mathbf{P}^{c,n}_{ij[s'd']}(t) * \mathbf{f}^n_{ij[s'd']}(t, \mathbf{X}^*_{ij}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ ij,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{i[sd]}(t) = -\nabla_{\mathbf{X}^k_{i[sd]}} g^c_{i}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'd']} \nabla_{\mathbf{X}^k_{i[sd]}} \mathbf{P}^{c,n}_{i[s'd']}(t) * \mathbf{f}^n_{i[s'd']}(t, \mathbf{X}^*_{i}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ i,\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{o[sd]}(t) = -\nabla_{\mathbf{X}^k_{o[sd]}} g^c_{[sd]}(t, \mathbf{X}^*_{o[sd]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \nabla_{\mathbf{X}^k_{o[sd]}} \mathbf{P}^{c,n}_{o[sd]}(t) * \mathbf{f}^n_{o[sd]}(t, \mathbf{X}^*_{o}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\dot{\mathbf{P}}^{c,k}_{[.d][sd]}(t) = -\nabla_{\mathbf{X}^k_{[.d][sd]}} g^c_{[.d]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) -$$

$$-\sum_n \sum_{[s'.]} \nabla_{\mathbf{X}^k_{[.d][sd]}} \mathbf{P}^{c,n}_{[.d][sd]}(t) * \mathbf{f}^n_{[.d][s'd]}(t, \mathbf{X}^*_{[.d]}, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\forall\ [sd] \in \mathbf{SD}^k,\ k$$

$$\sum_{[.d] \in \mathbf{D}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^c$$

$$\psi^{c*}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^c_{[s.]}$$

Theorem : Load Sharing

For each source, there must be flow only to destinations whose length is minimum:

$$\psi^{c*}_{[sd]}(t) > 0 \quad only \;\; if \qquad l^{c,Nash*}_{[sd]}(t) = \min_{[sd']}\{l^{c,Nash*}_{[sd']}(t)\}$$

$$\psi^{c*}_{[sd]}(t) = 0 \qquad\qquad o.w.$$

$$\sum_{[.d]\in \mathbf{S}^c_{[s.]}} \psi^{c*}_{[sd]}(t) = 1 \quad \forall\, [.d] \in \mathbf{D}^c_{[s.]},\;\; [s.] \in \mathbf{S}^c,\;\; c$$

and satisfies the partial differential vectors for the state and the costate variables.

So, in this section we have formulated and solved the dynamic join load sharing, routing and congestion control problem as a dynamic Nash game among multiple competing classes.

5.3 Stackelberg Equilibrium Solution

In this section, we formulate the dynamic join load sharing, routing and congestion control problem in distributed systems with two classes of jobs, one more powerful than the other, as a non-cooperative dynamic Stackelberg game. An example of such classes of jobs is when they have different priorities. Another example is when there is a system administrator (leader) and users (followers) with different objectives and power.

Customers of the most powerful class try to use the resources of the distributed system for their own benefit, ignoring the inconvenience that they cause to customers from the less powerful class.

Next, we briefly survey research on the dynamic Stackelberg game theory:

Starr & Ho [464] introduce nonzero-sum differential games and discuss Nash equilibrium, minimax and noninferior strategies. Then they solve the linear-quadratic game.

Chen & Cruz [96] analyze Stackelberg games with biased information. They present necessary conditions for open-loop strategies and use dynamic programming to define feedback strategies. Simaan & Cruz [449, 448] derive necessary and sufficient conditions for Stackelberg games. They also solve the linear-quadratic problem. Cruz [117] considers hierarchical games with multiple players at each level.

Basar & Selbuz [28, 29] consider linear-quadratic Stackelberg games. They derive a linear one-step memory closed-loop solution for the leader and a linear feedback solution for the follower. Basar [25] obtains the sufficient conditions for a three-player hierarchical game. Then he applies them to linear-quadratic games.

Papavassilopoulos & Cruz [376] analyze Stackelberg dynamic games, which are nonclassical control problems, since the control depends both on the state and time and its partial derivative with respect to the state appears in the state equation and in the cost function. They also [375] derive sufficient conditions for Stackelberg and Nash strategies for linear quadratic deterministic differential games when the players have memory.

In the following, we shall develop a methodology for the joint dynamic load sharing, routing and congestion control problem based on the Stackelberg game theory.

Next, we give some definitions for a two-hierarchical-class game similar to those in [27] for Stackelberg games:

Definition :

In a two class join load sharing, routing and congestion control problem, with the most powerful class α as the leader and the less powerful class β as the follower, the set $\mathcal{R}^\beta(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha)$, defined for the class α strategy $(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha) \in (\mathbf{RC}^\alpha, \mathbf{LS}^\alpha)$, by:

$$\begin{aligned} \mathcal{R}^\beta(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha) = \{ \quad & (\mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta) \in (\mathbf{RC}^\beta, \mathbf{\Psi}^\beta) \ \textit{such that}: \\ & J^\beta(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha, \mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta) \leq J^\beta(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha, \hat{\mathbf{\Phi}}^\beta, \hat{\mathbf{\Psi}}^\beta), \\ & \forall \ (\hat{\mathbf{\Phi}}^\beta, \hat{\mathbf{\Psi}}^\beta), \ \ \textit{such that} \ \ (\hat{\mathbf{\Phi}}^\beta, \hat{\mathbf{\Psi}}^\beta) \in (\mathbf{RC}^\beta, \mathbf{LS}^\beta)\} \end{aligned}$$

is the optimal response (rational reaction) set of the less powerful class β to the strategy of the most powerful class α.

What the above definition says is that the less powerful class β chooses its decision vector $(\mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta)$, that minimizes its cost function $J^\beta(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha, \mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta)$, for given strategy $(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha)$ of the most powerful class α.

Definition :

In a two class join load sharing, routing and congestion control problem with the most powerful class α as the leader, a strategy $(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*}) \in (\mathbf{RC}^\alpha, \mathbf{LS}^\alpha)$ is called a Stackelberg equilibrium strategy for the most powerful class α if and only if

$$\inf_{(\mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta) \in \mathcal{R}^\beta(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*})} J^\alpha(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*}, \mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta) \leq$$

$$\leq \inf_{(\mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta) \in \mathcal{R}^\beta(\mathbf{\Phi}^{\alpha}, \mathbf{\Psi}^{\alpha})} J^\alpha(\mathbf{\Phi}^{\alpha}, \mathbf{\Psi}^{\alpha}, \mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta) \quad \forall \ (\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha) \in (\mathbf{RC}^\alpha, \mathbf{LS}^\alpha)$$

This means that the most powerful class α chooses its strategy $(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*})$ that minimizes its cost function $J^\alpha(\mathbf{\Phi}^\alpha, \mathbf{\Psi}^\alpha, \mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta)$, given the optimal response set $\mathcal{R}^\beta(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*})$ of the less powerful class β to its strategy $(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*})$.

Definition :

Let $(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*}) \in (\mathbf{RC}^{\alpha}, \mathbf{LS}^{\alpha})$ *be a Stackelberg strategy for the most powerful class* α. *Then any element* $(\mathbf{\Phi}^{\beta*}, \mathbf{\Psi}^{\beta*}) \in \mathcal{R}^{\beta}(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*})$ *is an optimal strategy for the less powerful class* β *that is in equilibrium with* $(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*})$. *The strategy* $(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*}, \mathbf{\Phi}^{\beta*}, \mathbf{\Psi}^{\beta*})$ *is a* Stackelberg solution *for the game with the most powerful class* α *as the leader and the cost pair* $J^{\alpha}(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*}, \mathbf{\Phi}^{\beta*}, \mathbf{\Psi}^{\beta*})$, $J^{\beta}(\mathbf{\Phi}^{\alpha*}, \mathbf{\Psi}^{\alpha*}, \mathbf{\Phi}^{\beta*}, \mathbf{\Psi}^{\beta*})$ *is the corresponding Stackelberg equilibrium outcome.*

5.3.1 Optimal Control Formulation

In this section, we formulate the dynamic non-cooperative join load sharing, routing and congestion control problem as an Optimal Control Problem (OCP).

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with two hierarchical classes, with fixed initial time t_0 *and final time* t_f.

If for each class c, $H^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}(t))$ *is differentiable and convex in* $(\mathbf{X}, \mathbf{\Phi}^c, \mathbf{\Psi}^c)$ $\in (\mathbf{R}, \mathbf{RC}^c, \mathbf{LS}^c)$ $\quad \forall\, t \in [t_0, t_f]$, *for each fixed value of* $(\mathbf{\Phi}^k, \mathbf{\Psi}^k) \in (\mathbf{RC}^k, \mathbf{LS}^k)$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a Stackelberg equilibrium if and only if it solves the following Optimal Control Problem:*

$$
\begin{array}{ll}
\textit{minimize} & \displaystyle\int_{t_0}^{t_f} g^{\alpha}(t, \mathbf{X}(t), \mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))dt \\
\textit{with respect to} & (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))) \\
\textit{such that} & \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\
& \mathbf{X}(t_0) = \mathbf{X}_0 \\
& (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t)) \in (\mathbf{RC}^{\alpha}, \mathbf{LS}^{\alpha}) \\
& (\mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t)) \in (\mathbf{RC}^{\beta}, \mathbf{LS}^{\beta}) \\
& \displaystyle\int_{t_0}^{t_f} g^{\beta}(t, \mathbf{X}(t), \begin{array}{c}\mathbf{\Phi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t) \\ \mathbf{\Psi}^{\alpha}(t), \mathbf{\Psi}^{\beta}(t)\end{array})dt = \\
& = \displaystyle\min_{(\hat{\mathbf{\Phi}}^{\beta}(t), \hat{\mathbf{\Psi}}^{\beta}(t)) \in (\mathbf{RC}^{\beta}, \mathbf{LS}^{\beta})} \int_{t_0}^{t_f} g^{\beta}(t, \mathbf{X}(t), \begin{array}{c}\mathbf{\Phi}^{\alpha}(t), \hat{\mathbf{\Phi}}^{\beta}(t) \\ \mathbf{\Psi}^{\alpha}(t), \hat{\mathbf{\Psi}}^{\beta}(t)\end{array})dt
\end{array}
$$

Proof: It follows from the definition of the Stackelberg equilibrium.□

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with two hierarchical classes, with fixed initial time t_0 *and final time* t_f.

Let for each class c, $H^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}(t))$ *is differentiable and convex in* $(\mathbf{X}, \mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{R}, \mathbf{RC}^c, \mathbf{LS}^c) \quad \forall\, t \in [t_0, t_f]$, *for each fixed value of* $(\mathbf{\Phi}^k, \mathbf{\Psi}^k) \in (\mathbf{RC}^k, \mathbf{LS}^k)$.

If $(\hat{\mathbf{\Phi}}^*(t, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}_0)) = (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is an* <u>*open-loop Stackelberg equilibrium*</u> *and* $\{\mathbf{X}^*(t),\ t \in [t_0, t_f]\}$ *is the corresponding state trajectory, then* $\exists\ \mathbf{P}^c(t) : [t_0, t_f] \rightarrow \mathbf{R}^n,\ \forall\ c$ *continuous and piecewise continuously differentiable vector functions, such that* $\forall\ t \in [t_0, t_f]$:

$$\text{minimize} \quad \int_{t_0}^{t_f} g^{\alpha}(t, \mathbf{X}(t), \mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))dt$$

$$\text{with respect to} \quad (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t), \mathbf{Q}^{\beta}(t))$$

$$\text{such that} \quad \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$$

$$\mathbf{X}(t_0) = \mathbf{X}_0$$

$$\left[\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{o[sd]}} - Q^{\beta}_{[sd]}(t)\right] * \phi^{\beta}_{o[sd]}(t) = 0 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\left[\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{\pi[sd]}} - Q^{\beta}_{[sd]}(t)\right] * \phi^{\beta}_{\pi[sd]}(t) = 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\left[\frac{\partial H^{\beta}}{\partial \psi^{\beta}_{[sd]}} - Q^{\beta}_{[s.]}(t)\right] * \psi^{\beta}_{[sd]}(t) = 0 \quad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

$$\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{o[sd]}} - Q^{\beta}_{[sd]}(t) \geq 0 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{\pi[sd]}} - Q^{\beta}_{[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial H^{\beta}}{\partial \psi^{\beta}_{[sd]}} - Q^{\beta}_{[s.]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

$$\dot{\mathbf{P}}^{\beta}(t) = -\nabla_{\mathbf{X}} H^{\beta}(t, \mathbf{X}, \mathbf{\Phi}(t), \mathbf{\Psi}(t), \mathbf{P}^{\beta}(t))$$

$$\mathbf{P}^{\beta}(t_f) = \mathbf{0}$$

$$\phi^{\alpha}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^{\alpha}_{[sd]}} \phi^{\alpha}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^{\alpha}$$

$$\sum_{[.d] \in \mathbf{D}^{\alpha}_{[s.]}} \psi^{\alpha}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^{\alpha}$$

$$\phi^{\alpha}_{o[sd]}(t),\ \phi^{\alpha}_{\pi[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\alpha}_{[sd]},\ [sd] \in \mathbf{SD}^{\alpha}$$

$$\psi^{\alpha}_{[sd]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^{\alpha}_{[s.]},\ [s.] \in \mathbf{S}^{\alpha}$$

$$\phi^{\beta}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]}} \phi^{\beta}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\sum_{[.d] \in \mathbf{D}^{\beta}_{[s.]}} \psi^{\beta}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^{\beta}$$

$$\phi^{\beta}_{o[sd]}(t),\ \phi^{\beta}_{\pi[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\psi^{\beta}_{[sd]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

Proof: The Lagrangian for the less powerful class β is

$$L^{\beta} = H^{\beta} + \sum_{[sd] \in \mathbf{SD}^{\beta}} Q^{\beta}_{[sd]} * \left[1 - \phi^{\beta}_{o[sd]} - \sum_{\pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]}} \phi^{\beta}_{\pi[sd]} \right] + \sum_{[s.] \in \mathbf{S}^{\beta}} Q^{\beta}_{[s.]} * \left[1 - \sum_{[.d] \in \mathbf{D}^{\beta}_{[s.]}} \psi^{\beta}_{[sd]} \right]$$

with $\phi^{\beta}_{o[sd]},\ \phi^{\beta}_{\pi[sd]},\ \psi^{\beta}_{[sd]} \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$

Pontryagin's maximum principle necessary conditions are:

$$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))$$

$$\mathbf{X}^*(t_0) = \mathbf{X}_0$$

$$\frac{\partial L^{\beta *}}{\partial \phi^{\beta}_{o[sd]}} * \phi^{\beta *}_{o[sd]}(t) = 0 \;\Rightarrow\; \left[\frac{\partial H^{\beta *}}{\partial \phi^{\beta}_{o[sd]}} - Q^{\beta}_{[sd]}(t)\right] * \phi^{\beta *}_{o[sd]}(t) = 0 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial L^{\beta *}}{\partial \phi^{\beta}_{\pi[sd]}} * \phi^{\beta *}_{\pi[sd]}(t) = 0 \;\Rightarrow\; \left[\frac{\partial H^{\beta *}}{\partial \phi^{\beta}_{\pi[sd]}} - Q^{\beta}_{[sd]}(t)\right] * \phi^{\beta *}_{\pi[sd]}(t) = 0$$

$$\forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial L^{\beta *}}{\partial \psi^{\beta}_{[sd]}} * \psi^{\beta *}_{[sd]}(t) = 0 \;\Rightarrow\; \left[\frac{\partial H^{\beta *}}{\partial \psi^{\beta}_{[sd]}} - Q^{\beta}_{[s.]}(t)\right] * \psi^{\beta *}_{[sd]}(t) = 0$$

$$\forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

$$\frac{\partial L^{\beta *}}{\partial \phi^{\beta}_{o[sd]}} \geq 0 \;\Rightarrow\; \frac{\partial H^{\beta *}}{\partial \phi^{\beta}_{o[sd]}} - Q^{\beta}_{[sd]}(t) \geq 0 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial L^{\beta *}}{\partial \phi^{\beta}_{\pi[sd]}} \geq 0 \;\Rightarrow\; \frac{\partial H^{\beta *}}{\partial \phi^{\beta}_{\pi[sd]}} - Q^{\beta}_{[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial L^{\beta *}}{\partial \psi^{\beta}_{[sd]}} \geq 0 \;\Rightarrow\; \frac{\partial H^{\beta *}}{\partial \psi^{\beta}_{[sd]}} - Q^{\beta}_{[s.]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

$$\dot{\mathbf{P}}^{\beta}(t) = -\nabla_{\mathbf{X}} H^{\beta}(t, \mathbf{X}^*, \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}^{\beta}(t))$$

$$\mathbf{P}^{\beta}(t_f) = \mathbf{0}$$

$$\frac{\partial L^{\beta *}}{\partial Q^{\beta}_{[sd]}} = 0 \Rightarrow \phi^{\beta *}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]}} \phi^{\beta *}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial L^{\beta *}}{\partial Q^{\beta}_{[s.]}} = 0 \Rightarrow \sum_{[.d] \in \mathbf{D}^{\beta}_{[s.]}} \psi^{\beta *}_{[sd]}(t) = 1 \quad \forall\ [s.] \in \mathbf{S}^{\beta}$$

$$\phi^{\beta *}_{o[sd]}(t),\ \phi^{\beta *}_{\pi[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\psi^{\beta *}_{[sd]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

□

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with two hierarchical classes, with fixed initial time t_0 *and final time* t_f.

Let for each class c, $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, *are continuously differentiable with respect to* $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{RC}, \mathbf{LS})$, $\forall\ t \in [t_0, t_f]$.

If $(\hat{\mathbf{\Phi}}^*(t, \mathbf{X}, \mathbf{X}_0), \hat{\mathbf{\Psi}}^*(t, \mathbf{X}, \mathbf{X}_0)) = (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ *is a closed-loop memoryless Stackelberg equilibrium such that* $(\hat{\mathbf{\Phi}}^{c*}(t, \mathbf{X}, \mathbf{X}_0), \hat{\mathbf{\Phi}}^{c*}(t, \mathbf{X}, \mathbf{X}_0))$ *is continuously differentiable with respect to* $\mathbf{X} \in \mathbf{R}^n$, $\forall\ c$, $t \in [t_0, t_f]$ *and* $\{\mathbf{X}^*(t),\ t \in [t_0, t_f]\}$ *is the corresponding state trajectory, then* $\exists\ \mathbf{P}^c(t) : [t_0, t_f] \to \mathbf{R}^n$, $\forall\ c$, *continuous and piecewise continuously differentiable vector functions, such that* $\forall\ t \in [t_0, t_f]$:

minimize $\int_{t_0}^{t_f} g^{\alpha}(t, \mathbf{X}(t), \mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))dt$

with respect to $(\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t), \mathbf{Q}^{\beta}(t))$

such that $\dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$

$$\mathbf{X}(t_0) = \mathbf{X}_0$$

$$\left[\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{o[sd]}} - Q^{\beta}_{[sd]}(t)\right] * \phi^{\beta}_{o[sd]}(t) = 0 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\left[\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{\pi[sd]}} - Q^{\beta}_{[sd]}(t)\right] * \phi^{\beta}_{\pi[sd]}(t) = 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\left[\frac{\partial H^{\beta}}{\partial \psi^{\beta}_{[sd]}} - Q^{\beta}_{[s.]}(t)\right] * \psi^{\beta}_{[sd]}(t) = 0 \quad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

$$\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{o[sd]}} - Q^{\beta}_{[sd]}(t) \geq 0 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{\pi[sd]}} - Q^{\beta}_{[sd]}(t) \geq 0 \quad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\frac{\partial H^{\beta}}{\partial \psi^{\beta}_{[sd]}} - Q^{\beta}_{[s.]}(t) \geq 0 \quad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

$$\dot{\mathbf{P}}^{\beta}(t) = -\nabla_{\mathbf{X}} H^{\beta}(t, \mathbf{X}, \hat{\mathbf{\Phi}}(t, \mathbf{X}, \mathbf{X}_0), \hat{\mathbf{\Psi}}(t), \mathbf{X}, \mathbf{X}_0), \mathbf{P}^{\beta}(t))$$

$$\mathbf{P}^{\beta}(t_f) = \mathbf{0}$$

$$\phi^{\alpha}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^{\alpha}_{[sd]}} \phi^{\alpha}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^{\alpha}$$

$$\sum_{[.d] \in \mathbf{D}^{\alpha}_{[s.]}} \psi^{\alpha}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^{\alpha}$$

$$\phi^{\alpha}_{o[sd]}(t),\ \phi^{\alpha}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^{\alpha}_{[sd]},\ [sd] \in \mathbf{SD}^{\alpha}$$

$$\psi^{\alpha}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^{\alpha}_{[s.]},\ [s.] \in \mathbf{S}^{\alpha}$$

$$\phi^{\beta}_{o[sd]}(t) + \sum_{\pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]}} \phi^{\beta}_{\pi[sd]}(t) = 1 \quad \forall\ [sd] \in \mathbf{SD}^{\beta}$$

$$\sum_{[.d] \in \mathbf{D}^{\beta}_{[s.]}} \psi^{\beta}_{[sd]}(t) = 1 \qquad \forall\ [s.] \in \mathbf{S}^{\beta}$$

$$\phi^{\beta}_{o[sd]}(t),\ \phi^{\beta}_{\pi[sd]}(t) \geq 0 \qquad \forall\ \pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]},\ [sd] \in \mathbf{SD}^{\beta}$$

$$\psi^{\beta}_{[sd]}(t) \geq 0 \qquad \forall\ [.d] \in \mathbf{D}^{\beta}_{[s.]},\ [s.] \in \mathbf{S}^{\beta}$$

Proof: The proof is similar to that for the open-loop Stackelberg equilibrium.□

One of the disadvantages of using Stackelberg strategies is that the principle of optimality does not hold for the leader. A modification of the Stackelberg strategy concept requires that the strategies for the remaining time-to-go after each stage should be optimal.

5.3.2 Nonlinear Complementarity Problem Formulation

In this section, we formulate the dynamic two hierarchical class load sharing, routing and congestion control problem as a Nonlinear Complementarity Problem (NCP).

Define the vector of classβ congestion control, routing and load sharing fractions as well as Lagrange multipliers:

$$\mathbf{Z}^{\beta}(t) = \left[\ldots \ \phi^{\beta}_{o[sd]}(t) \ \ldots \ \phi^{\beta}_{\pi[sd]} \ \ldots \ Q^{\beta}_{[sd]}(t) \ \ldots \ \psi^{\beta}_{[sd]}(t) \ \ldots \ Q^{\beta}_{[s.]}(t) \ \ldots \right]^{T}$$

and the vector of class β derivative of its Lagrangian with respect to the congestion control, routing and load sharing fractions as well as Lagrange multipliers:

$$\begin{aligned} \nabla L^{\beta}(\mathbf{Z}(t)) = & \left[\ldots \left(\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{o[sd]}} - Q^{\beta}_{o[sd]}(t) \right) \ldots \left(\frac{\partial H^{\beta}}{\partial \phi^{\beta}_{\pi[sd]}} - Q^{\beta}_{\pi[sd]}(t) \right) \ldots \right. \\ & \ldots \left(1 - \phi^{\beta}_{o[sd]}(t) - \sum_{\pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]}} \phi^{\beta}_{\pi[sd]}(t) \right) \ldots \\ & \left. \ldots \left(\frac{\partial H^{\beta}}{\partial \psi^{\beta}_{[sd]}} - Q^{\beta}_{[s.]}(t) \right) \ldots \left(1 - \sum_{[.d] \in \mathbf{D}^{\beta}_{[s.]}} \psi^{\beta}_{[sd]}(t) \right) \ldots \right] \end{aligned}$$

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with two hierarchical classes, with fixed initial time t_0 and final time t_f.

If for each class c, $H^{c}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}, \mathbf{P}(t))$ is differentiable and convex in $(\mathbf{X}, \mathbf{\Phi}^{c}, \mathbf{\Psi}^{c})$ $\in (\mathbf{R}, \mathbf{RC}^{c}, \mathbf{LS}^{c})$ $\quad \forall\, t \in [t_0, t_f]$, for each fixed value of $(\mathbf{\Phi}^{k}, \mathbf{\Psi}^{k}) \in (\mathbf{RC}^{k}, \mathbf{LS}^{k})$,

then $(\mathbf{\Phi}^{*}(t), \mathbf{\Psi}^{*}(t)) \in (\mathbf{RC}, \mathbf{LS})$ is a Stackelberg equilibrium if and only if it solves the following problem $\forall\, t \in [t_0, t_f]$:

$$
\begin{aligned}
&\textit{minimize} && \int_{t_0}^{t_f} g^{\alpha}(t, \mathbf{X}(t), \mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))dt \\
&\textit{with respect to} && (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))) \\
&\textit{such that} && \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\
& && \mathbf{X}(t_0) = \mathbf{X}_0 \\
& && (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t)) \in (\mathbf{RC}^{\alpha}, \mathbf{LS}^{\alpha}) \\
& && (\mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t)) \in (\mathbf{RC}^{\beta^{\beta}}, \mathbf{LS}^{\beta}) \\
& && \nabla L^{\beta}(\mathbf{Z}^{\beta *}(t)) * \mathbf{Z}^{\beta *}(t) = 0 \\
& && \nabla L^{\beta}(\mathbf{Z}^{\beta *}(t)) \geq 0 \\
& && \mathbf{Z}^{\beta *}(t) \geq 0 \\
& && \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\
& && \mathbf{X}(t_0) = \mathbf{X}_0 \\
& && \dot{\mathbf{P}}^{\beta}(t) = -\nabla_{\mathbf{X}} H^{\beta}(t, \mathbf{X}, \mathbf{\Phi}(t), \mathbf{\Psi}(t), \mathbf{P}^{\beta}(t)) \\
& && \mathbf{P}^{\beta}(t_f) = \mathbf{0}
\end{aligned}
$$

Proof: After some algebraic manipulations, we find that the NCP: $\nabla L(\mathbf{Z}(t)) * \mathbf{Z}(t) = 0$; $\nabla L(\mathbf{Z}(t)) \geq 0$; $\mathbf{Z}(t) \geq 0$ with $\mathbf{Z}(t)$ and $\nabla L(\mathbf{Z}(t))$ as defined above, is equivalent to the Pontryagin's maximum principle necessary conditions for the follower. □

5.3.3 Variational Inequality Formulation

In this section, we formulate the dynamic non-cooperative load sharing, routing and congestion control problem as a Variational Inequality Problem (VIP).

Define the vector of class β congestion control, routing and load sharing fractions:

$$(\mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t)) = \left[\ldots\ \phi^{\beta}_{o[sd]}(t)\ \ldots\ \phi^{\beta}_{\pi[sd]}(t)\ \ldots\ \psi^{\beta}_{[sd]}(t)\ \ldots \right]^{T}$$

as well the vector of class β derivatives of its Lagrangian with respect to the congestion control, routing and load sharing fractions:

$$\nabla H^{\beta}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t), \mathbf{P}(t)) \quad = \quad \left[\ldots \frac{\partial H^{\beta}}{\partial \phi^{\beta}_{o[sd]}} \ldots \sum_{\pi[sd] \in \mathbf{\Pi}^{\beta}_{[sd]}} \frac{\partial H^{\beta}}{\partial \phi^{\beta}_{\pi[sd]}} \ldots \frac{\partial H^{\beta}}{\partial \psi^{\beta}_{[sd]}} \ldots \right]$$

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with two hierarchical classes, with fixed initial time t_0 and final time t_f.

Let for each class c, $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, be continuously differentiable with respect to $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi})$ $\forall\, t \in [t_0, t_f]$. If H^c is continuously differentiable and convex in $(\mathbf{X}, \mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{R}^n, \mathbf{RC}^c, \mathbf{LS}^c)$, $\forall\, t \in [t_0, t_f]$, for each fixed value of $(\mathbf{\Phi}^k(t), \mathbf{\Psi}^k(t)) \in (\mathbf{RC}^k, \mathbf{LS}^k)$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ is a Stackelberg equilibrium if and only if it solves the following problem $\forall\, t \in [t_0, t_f]$:

$$\text{minimize} \qquad \int_{t_0}^{t_f} g^{\alpha}(t, \mathbf{X}(t), \mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))dt$$

$$\text{with respect to} \quad (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t)))$$

$$\text{such that} \qquad \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$$

$$\mathbf{X}(t_0) = \mathbf{X}_0$$

$$(\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t)) \in (\mathbf{RC}^{\alpha}, \mathbf{LS}^{\alpha})$$

$$(\mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t)) \in (\mathbf{RC}^{\beta^{\beta}}, \mathbf{LS}^{\beta})$$

$$\nabla H^{\beta}(t, \mathbf{X}^*(t), \mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t), \mathbf{P}(t)) * ((\mathbf{\Phi}, \mathbf{\Psi}) - (\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t))) \geq 0$$
$$\forall \ (\mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{RC}, \mathbf{LS})$$

$$\dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t))$$

$$\mathbf{X}(t_0) = \mathbf{X}_0$$

$$\dot{\mathbf{P}}^{\beta}(t) = -\nabla_{\mathbf{X}} H^{\beta}(t, \mathbf{X}, \mathbf{\Phi}(t), \mathbf{\Psi}(t), \mathbf{P}^{\beta}(t))$$

$$\mathbf{P}^{\beta}(t_f) = \mathbf{0}$$

Proof: If $(\mathbf{\Phi}^{\beta*}(t), \mathbf{\Psi}^{\beta*}(t))$ is a local minimum for the following minimization problem

$$
\begin{array}{ll}
\textit{minimize} & \displaystyle\int_{t_0}^{t_f} g^\beta(t, \mathbf{X}(t), \begin{array}{c}\mathbf{\Phi}^{\alpha*}(t), \mathbf{\Phi}^\beta(t)\\ \mathbf{\Psi}^{\alpha*}(t), \mathbf{\Psi}^\beta(t)\end{array})dt \\
\textit{with respect to} & (\mathbf{\Phi}^\beta(t), \mathbf{\Psi}^\beta(t)) \\
\textit{such that} & \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\
 & \mathbf{X}(t_0) = \mathbf{X}_0 \\
 & \\
 & (\mathbf{\Phi}^\beta(t), \mathbf{\Psi}^\beta(t)) \in (\mathbf{RC}^\beta, \mathbf{LS}^\beta)
\end{array}
$$

and g^β is a continuously differentiable convex function over the nonempty convex, closed and bounded set $(\mathbf{RC}^\beta, \mathbf{LS}^\beta)$, then $\forall\ t \in [t_0, t_f]$:

$$
\sum_{[sd]\in\mathbf{SD}^\beta} \left\{ \frac{\partial H^{\beta*}}{\partial \phi^\beta_{o[sd]}} * (\phi^\beta_{o[sd]} - \phi^{\beta*}_{o[sd]}(t)) + \right.
$$

$$
+ \sum_{\pi[sd]\in\mathbf{\Pi}^\beta_{[sd]}} \frac{\partial H^{\beta*}}{\partial \phi^\beta_{\pi[sd]}} * (\phi^\beta_{\pi[sd]} - \phi^{\beta*}_{\pi[sd]}(t)) +
$$

$$
\left. + \frac{\partial H^{\beta*}}{\partial \psi^\beta_{[sd]}} * (\psi^\beta_{[sd]} - \psi^{\beta*}_{[sd]}(t)) \right\} \geq 0 \quad \forall\ (\mathbf{\Phi}^\beta, \mathbf{\Psi}^\beta) \in (\mathbf{RC}^\beta, \mathbf{LS}^\beta)
$$

□

Another equivalent formulation is the following Theorem:

Theorem :

Consider the dynamic join load sharing, routing and congestion control problem in distributed systems with two hierarchical classes, with fixed initial time t_0 and final time t_f.

Let for each class c, $g^c(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, $\mathbf{f}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$, be continuously differentiable with respect to $(\mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi}) \in (\mathbf{R}^n, \mathbf{\Phi}, \mathbf{\Psi})$ $\forall\ t \in [t_0, t_f]$. If H^c is continuously differentiable and convex in $(\mathbf{X}, \mathbf{\Phi}^c, \mathbf{\Psi}^c) \in (\mathbf{R}^n, \mathbf{RC}^c, \mathbf{LS}^c)$, $\forall\ t \in [t_0, t_f]$, for each fixed value of $(\mathbf{\Phi}^k(t), \mathbf{\Psi}^k(t)) \in (\mathbf{RC}^k, \mathbf{LS}^k)$,

then $(\mathbf{\Phi}^*(t), \mathbf{\Psi}^*(t)) \in (\mathbf{RC}, \mathbf{LS})$ is a Stackelberg equilibrium if and only if it solves the following problem $\forall\ t \in [t_0, t_f]$:

$$\begin{array}{ll}
\textit{minimize} & \int_{t_0}^{t_f} g^{\alpha}(t, \mathbf{X}(t), \mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))dt \\
\textit{with respect to} & (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t))) \\
\textit{such that} & \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\
& \mathbf{X}(t_0) = \mathbf{X}_0 \\
& (\mathbf{\Phi}^{\alpha}(t), \mathbf{\Psi}^{\alpha}(t)) \in (\mathbf{RC}^{\alpha}, \mathbf{LS}^{\alpha}) \\
& (\mathbf{\Phi}^{\beta}(t), \mathbf{\Psi}^{\beta}(t)) \in (\mathbf{RC}^{\beta\beta}, \mathbf{LS}^{\beta}) \\
& \int_{t_0}^{t_f} g^{\beta}(t, \mathbf{X}(t), \begin{matrix} \mathbf{\Phi}^{\alpha}(t), \mathbf{\Phi}^{\beta}(t) \\ \mathbf{\Psi}^{\alpha}(t), \mathbf{\Psi}^{\beta}(t) \end{matrix})dt = \\
& = \min_{(\hat{\mathbf{\Phi}}^{\beta}(t), \hat{\mathbf{\Psi}}^{\beta}(t)) \in (\mathbf{RC}^{\beta\beta}, \mathbf{LS}^{\beta})} \int_{t_0}^{t_f} g^{\beta}(t, \mathbf{X}(t), \begin{matrix} \mathbf{\Phi}^{\alpha}(t), \hat{\mathbf{\Phi}}^{\beta}(t) \\ \mathbf{\Psi}^{\alpha}(t), \hat{\mathbf{\Psi}}^{\beta}(t) \end{matrix})dt \\
& \nabla L^{\beta}(\mathbf{Z}^{\beta}(t)^{*}) * (\mathbf{Z}^{\beta} - \mathbf{Z}^{\beta}(t)^{*}) \geq 0 \qquad \forall\ \mathbf{Z}^{\beta} > 0 \\
& \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) \\
& \mathbf{X}(t_0) = \mathbf{X}_0 \\
& \dot{\mathbf{P}}^{\beta}(t) = -\nabla_{\mathbf{X}} H^{\beta}(t, \mathbf{X}, \mathbf{\Phi}(t), \mathbf{\Psi}(t), \mathbf{P}^{\beta}(t)) \\
& \mathbf{P}^{\beta}(t_f) = \mathbf{0}
\end{array}$$

Proof: The NCP: $f(x^{*}) * x^{*} = 0 \;\; f(x^{*}) \geq 0 \;\; x^{*} > 0$
and the VIP: find x^{*} such that $f(x^{*}) * (x - x^{*}) \geq 0 \;\; \forall x > 0$
are equivalent.□

5.3.4 Maximum Principle for Separable Cost Functions

In this section, we derive the first order necessary and sufficient conditions for a Stackelberg equilibrium on the path flows, when the cost function at each resource depends only on the flow on this resource.

The partial derivatives of the cost function $g^{\beta}(t, \mathbf{X}, \mathbf{\Phi}, \mathbf{\Psi})$ with respect to the path fractions $\phi^{\beta}_{\pi[sd]}$ can be written with respect to the link flows λ^{β}_{ij} and node flows λ^{β}_{i}:

$$\frac{\partial g^{\beta}_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^{\beta}_{\pi[sd]}} = \frac{\partial g^{\beta}_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda^{\beta}_{ij}} * \frac{\partial \lambda^{\beta}_{ij}}{\partial \phi^{\beta}_{\pi[sd]}} =$$

$$= \frac{\partial g^{\beta}_{ij}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda^{\beta}_{ij}} * (\gamma^{\beta}_{[sd]}(t) + \gamma^{\beta}_{[s.]}(t) * \psi^{\beta}_{[sd]}) * 1_{ij \in \pi[sd]}(t)$$

$$\frac{\partial g^{\beta}_{i}(t, \mathbf{X}_{i}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^{\beta}_{\pi[sd]}} = \frac{\partial g^{\beta}_{i}(t, \mathbf{X}_{i}, \mathbf{\Lambda}_{i})}{\partial \lambda^{\beta}_{i}} * \frac{\partial \lambda^{\beta}_{i}}{\partial \phi^{\beta}_{\pi[sd]}} =$$

$$= \frac{\partial g^{\beta}_{i}(t, \mathbf{X}_{i}, \mathbf{\Lambda}_{i})}{\partial \lambda^{\beta}_{i}} * (\gamma^{\beta}_{[sd]}(t) + \gamma^{\beta}_{[s.]}(t) * \psi^{\beta}_{[sd]}) * 1_{i \in \pi[sd]}(t)$$

$$\frac{\partial g^{\beta}_{o[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \phi^{\beta}_{o[sd]}} = \frac{\partial g^{\beta}_{o[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda^{\beta}_{o[sd]}} * \frac{\partial \lambda^{\beta}_{o[sd]}}{\partial \phi^{\beta}_{o[sd]}} =$$

$$= \frac{\partial g^{\beta}_{o[sd]}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda^{\beta}_{o[sd]}} * (\gamma^{\beta}_{[sd]}(t) + \gamma^{\beta}_{[s.]}(t) * \psi^{\beta}_{[sd]})$$

$$\frac{\partial g_{ij}^{\beta}(t, \mathbf{X}_{ij}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^{\beta}} = \frac{\partial g_{ij}^{\beta}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda_{ij}^{\beta}} * \frac{\partial \lambda_{ij}^{\beta}}{\partial \psi_{[sd]}^{\beta}} =$$

$$= \sum_{\pi[sd] \in \mathbf{\Pi}_{[sd]}^{\beta}} \frac{\partial g_{ij}^{\beta}(t, \mathbf{X}_{ij}, \mathbf{\Lambda}_{ij})}{\partial \lambda_{ij}^{\beta}} * \gamma_{[s.]}^{\beta}(t) * \phi_{\pi[sd]}^{c*} * 1_{ij \in \pi[sd]}(t)$$

$$\frac{\partial g_{i}^{\beta}(t, \mathbf{X}_{i}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^{\beta}} = \frac{\partial g_{i}^{\beta}(t, \mathbf{X}_{i}, \mathbf{\Lambda}_{i})}{\partial \lambda_{i}^{\beta}} * \frac{\partial \lambda_{i}^{\beta}}{\partial \psi_{[sd]}^{\beta}} =$$

$$= \sum_{\pi[sd] \in \mathbf{\Pi}_{[sd]}^{\beta}} \frac{\partial g_{i}^{\beta}(t, \mathbf{X}_{i}, \mathbf{\Lambda}_{i})}{\partial \lambda_{i}^{\beta}} * \gamma_{[s.]}^{\beta}(t) * \phi_{\pi[sd]}^{c*} * 1_{i \in \pi[sd]}(t)$$

$$\frac{\partial g_{o[sd]}^{\beta}(t, \mathbf{X}_{o[sd]}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^{\beta}} = \frac{\partial g_{o[sd]}^{\beta}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda_{o[sd]}^{\beta}} * \frac{\partial \lambda_{o[sd]}^{\beta}}{\partial \psi_{[sd]}^{\beta}} =$$

$$= \frac{\partial g_{o[sd]}^{\beta}(t, \mathbf{X}_{o[sd]}, \mathbf{\Lambda}_{o[sd]})}{\partial \lambda_{o[sd]}^{\beta}} * \gamma_{[s.]}^{\beta}(t) * \phi_{o[sd]}^{c*}$$

$$\frac{\partial g_{[.d]}^{\beta}(t, \mathbf{X}_{[.d]}, \mathbf{\Phi}, \mathbf{\Psi})}{\partial \psi_{[sd]}^{\beta}} = \frac{\partial g_{[.d]}^{\beta}(t, \mathbf{X}_{[.d]}, \mathbf{\Lambda}_{[.d]})}{\partial \lambda_{[.d]}^{\beta}} * \frac{\partial \lambda_{[.d]}^{\beta}}{\partial \psi_{[sd]}^{\beta}} =$$

$$= \frac{\partial g_{[.d]}^{\beta}(t, \mathbf{X}_{[.d]}, \mathbf{\Lambda}_{[.d]})}{\partial \lambda_{[.d]}^{\beta}} * \gamma_{[s.]}^{\beta}(t)$$

5.4 Application to Datagram Networks

In this section, we apply the methodologies developed in the previous sections to datagram networks. We develop dynamic queueing models for the average number of class c packets in the queue and in the system (queue plus service) for multiple classes and priority classes $M/G/1$ queues. We also introduce the idea of using linearized approximate dynamic queueing models, in order to have a linear-quadratic problem for which there is extensive literature. We also suggest using second order dynamic queueing models, when the traffic can not be described only by first order models. Furthermore, we introduce Wiener process models for modeling the stochastic system. Finally, we present some cost functions and state constraints that can be used in the optimal control problem.

5.4.1 Dynamic Queueing Models for Multiple Classes

The general structure of the dynamic model introduced in this section is that the number of packets in a resource increases by the number of arrivals to and decreases by the number of departures from that resource. The departure rate should be a nonnegative, nondecreasing, continuous and concave function $\mu C * \rho(N)$ of the number of packets in the resource, with $\mu C * \rho(N) < N$. A dynamic model for $M/M/1$ queues, that was originally proposed by Agnew [4] and Rider [398] and was later used in network optimization studies by Filipiak [158, 159, 153], Economides, Ioannou & Silvester [137], Tipper & Sundareshan [485], is the following:

$$\dot{N}(t) = \lambda(t) - \mu C(t) * \frac{N(t)}{1 + N(t)}$$

Filipiak has also proposed a dynamic model for $M/M/\infty$ queues:

$$\dot{N}(t) = \lambda(t) - \mu C(t) * N(t)$$

as well as for $M/D/1$ queues:

$$\dot{N}(t) = \lambda(t) - \mu C(t) * \left(1 + N(t) - \sqrt{1 + (N(t))^2}\right)$$

Next, we extend the above models for multiple class $M/G/1$ queues. The average number of class c packets in $M/G/1$ queues is given by

$$N^c = \rho^c + \rho^c * \frac{\rho * \overline{x^2} * \mu^2}{2(1-\rho)} \quad \forall\ c$$

where ρ^c is the utilization for class c and $\rho = \sum_c \rho^c$ is the overall utilization.

Solving the above system of equations for ρ^c, we have the utilization for class c as a function of the average number of packets for each class:

$$\rho^c = \frac{2N^c * \left(1 - \overline{x^2} * \mu^2 - \sum_k N^k + \sqrt{\left(1 + \sum_k N^k\right)^2 - 2\sum_k N^k * (2 - \overline{x^2} * \mu^2)}\right)}{(2 - \overline{x^2} * \mu^2) * \left(1 - \sum_k N^k + \sqrt{\left(1 + \sum_k N^k\right)^2 - 2\sum_k N^k * (2 - \overline{x^2} * \mu^2)}\right)}$$

Then we propose the following dynamic model for multiple class $M/G/1$ queues:

$$\dot{N}^c(t) = \lambda^c(t) - \mu C(t) * \frac{2N^c(t)}{2 - \overline{x^2} * \mu^2} *$$

$$* \frac{\left(1 - \overline{x^2} * \mu^2 - \sum_k N^k(t) + \sqrt{\left(1 + \sum_k N^k(t)\right)^2 - 2\sum_k N^k(t) * (2 - \overline{x^2} * \mu^2)}\right)}{\left(1 - \sum_k N^k(t) + \sqrt{\left(1 + \sum_k N^k(t)\right)^2 - 2\sum_k N^k(t) * (2 - \overline{x^2} * \mu^2)}\right)}$$

For exponential service, general service and Processor Sharing ($P.S.$) discipline and deterministic service times, the above model gives the following dynamic models:

$$\dot{N}^c(t) = \lambda^c(t) - \mu C(t) * \frac{N^c(t)}{1 + \sum_k N^k(t)} \qquad M/M/1$$

$$\dot{N}^c(t) = \lambda^c(t) - \mu C(t) * \frac{w^c * N^c(t)}{1 + \sum_k w^k * N^k(t)} \qquad \textit{class discriminating P.S.}$$

$$\dot{N}^c(t) = \lambda^c(t) - \mu C(t) * \frac{2N^c(t) * \left(-\sum_k N^k(t) + \sqrt{1 + \left(\sum_k N^k(t) \right)^2} \right)}{1 - \sum_k N^k(t) + \sqrt{1 + \left(\sum_k N^k(t) \right)^2}} \quad M/D/1$$

Also, for multiple class $M/M/\infty$ queues we have the following dynamic model:

$$\dot{N}^c(t) = \lambda^c(t) - \mu^c C(t) * N^c(t) \quad M/M/\infty$$

5.4.2 Linearized Dynamic Queueing Models

Although the above dynamic queueing models describe accurately the dynamic behavior of the queue, they depend nonlinearly on the average number of packets in the system (except the $M/M/\infty$ model). Therefore the analytical solution of the dynamic optimization problem usually becomes intractable. Next, we propose the linearization of the above dynamic queueing models, that gives simpler models. For example, the linearized multiple class $M/M/1$ queueing model is the following:

$$
\begin{aligned}
\dot{N}^c(t) &= \lambda^c(t) - \mu C * \frac{N^c(t)}{1 + \sum\limits_k N^k(t)} \\
&\approx \lambda^c(t) - \mu C * \frac{\overline{N^c}}{1 + \sum\limits_k \overline{N^k}} - \mu C * \sum_k \frac{\partial}{\partial \overline{N^k}} \left(\frac{\overline{N^c}}{1 + \sum\limits_k \overline{N^k}} \right) * (N^k(t) - \overline{N^k}) \\
&\approx \lambda^c(t) - \mu C * \frac{\overline{N^c}}{1 + \sum\limits_k \overline{N^k}} - \mu C * \frac{1 + \sum\limits_{k \neq c} \overline{N^k}}{\left(1 + \sum\limits_k \overline{N^k}\right)^2} * (N^c(t) - \overline{N^c}) + \\
&\quad + \mu C * \sum_k \frac{N^c}{\left(1 + \sum\limits_n \overline{N^n}\right)^2} * (N^k(t) - \overline{N^k}) \\
&\approx \lambda^c(t) - \mu C * \frac{\dfrac{\lambda^c}{\mu C - \sum\limits_k \lambda^k}}{1 + \dfrac{\sum\limits_k \lambda^k}{\mu C - \sum\limits_k \lambda^k}} - \\
&\quad - \mu C * \frac{1}{1 + \dfrac{\sum\limits_k \lambda^k}{\mu C - \sum\limits_k \lambda^k}} * \left(N^c(t) - \frac{\lambda^c}{\mu C - \sum\limits_k \lambda^k} \right) + \\
&\quad + \mu C * \frac{\lambda^c}{\mu C \sum\limits_n \lambda^n} * \sum_k \frac{1}{\left(1 + \dfrac{\sum\limits_n \lambda^n}{\mu C - \sum\limits_n \lambda^n}\right)^2} * \left(N^k(t) - \frac{\lambda^k}{\mu C - \sum\limits_n \lambda^n} \right)
\end{aligned}
$$

Finally, we have the following linearized model for multi-class $M/M/1$ queues:

$$\dot{N}^c(t) \approx \lambda^c(t) - \lambda^c * \frac{\sum_k \lambda^k}{\mu C} - (\mu C - \sum_k \lambda^k) * N^c(t) + \frac{\lambda^c * \mu C - \sum_n \lambda^n}{\mu C} * \sum_k N^k(t)$$

The above model satisfies the steady-state flow conservation

$$\lambda^c - \lambda^c * \frac{\sum_k \lambda^k}{\mu C} = (\mu C - \sum_k \lambda^k) * \overline{N^c} - \frac{\lambda^c * \mu C - \sum_n \lambda^n}{\mu C} * \sum_k \overline{N^k} \Leftrightarrow \overline{N^c} = \frac{\lambda^c}{\mu C - \sum_k \lambda^k}$$

Similarly, we may derive dynamic models for the average number of customers in the systems (queue plus service), or in the queue, for multiple class, priority class $M/G/1$ queues.

Another approximate model for multiple class $M/M/1$ queues is the following:

$$\begin{aligned}
\dot{N}^c(t) &= \lambda^c(t) - \mu C * \frac{N^c(t)}{1 + \sum_k N^k(t)} \\
&\approx \lambda^c(t) - \mu C * \frac{1}{1 + \sum_k \overline{N^k}} * N^c(t) \\
&\approx \lambda^c(t) - \mu C * \frac{1}{1 + \frac{\sum_k \lambda^k}{\mu C(t) - \sum_k \lambda^k}} * N^c(t) \\
&\approx \lambda^c(t) - (\mu C - \sum_k \lambda^k) * N^c(t)
\end{aligned}$$

The above model satisfies the steady-state flow conservation

$$\lambda^c = (\mu C - \sum_k \lambda^k) * N^c \Leftrightarrow N^c = \frac{\lambda}{\mu C - \sum_k \lambda^k}$$

The link length becomes

$$l = \left[\frac{\partial}{\partial N^c}\left((\mu C - \sum_k \lambda^k) * N^c\right)\right]^{-1} = \frac{1}{\mu C - \sum_k \lambda^k} = \frac{\frac{1}{\mu C}}{1 - \frac{\sum_k \lambda^k}{\mu C}} =$$

$$= \frac{\frac{1}{\mu C}}{1 - \frac{\sum_k N^k}{1 + \sum_k N^k}} = \frac{1 + \sum_k N^k}{\mu C}$$

This result explains why the shortest route routing achieves good performance (see section 5.6.5).

After the model linearization, the system state is described by the following state equation

$$\dot{\mathbf{X}} = \mathbf{A} * \mathbf{X} + \mathbf{B} * \mathbf{U} \qquad \mathbf{X}_0 : given$$

with cost function

$$\int_{t_0}^{t_f} \frac{1}{2} * (\mathbf{X}^T * \mathbf{Q} * \mathbf{X} + \mathbf{U}^T * \mathbf{R} * \mathbf{U}) dt$$

where A, B, Q, R are suitable matrices. Thus, we can use results from the optimal control theory on linear-quadratic problems, to solve the joint load sharing, routing and congestion control problem.

5.4.3 Dynamic Queueing Models for the Packets in Queue

In future high speed networks, we will have information only about the average number of packets in the queue (not both in the queue and in service), due to the enormous number of packets that will be in transit into the network. Therefore, it is also useful to have dynamic queueing models with state the average number of packets in the queue.

Here, we introduce a dynamic queueing model for the average number of packet in the multiple class $M/G/1$ queue. The average number of class c packets in queue for a multiple class $M/G/1$ queue is given by

$$N_Q^c = \rho^c * \frac{\rho * \overline{x^2} * \mu^2}{2(1-\rho)} \cdot \forall\ c$$

Solving the above system of equations, we have the utilization for class c, ρ^c, as a function of the average number of packets in queue for all classes

$$\rho^c = \frac{2N_Q^c * \left(\overline{x^2} * \mu^2 + \sum_k N_Q^k - \sqrt{\left(\sum_k N_Q^k\right)^2 + 2\sum_k N^Q * \overline{x^2} * \mu^2}\right)}{\overline{x^2} * \mu^2 * \left(-\sum_k N_Q^k + \sqrt{\left(\sum_k N_Q^k\right)^2 + 2\sum_k N^Q * \overline{x^2} * \mu^2}\right)}$$

Then we propose the following dynamic model for multiple class $M/G/1$ queues:

$$\dot{N}_Q^c(t) = \lambda^c(t) - \mu C(t) * \frac{2N_Q^c(t)}{\overline{x^2} * \mu^2} *$$

$$* \frac{\overline{x^2} * \mu^2 + \sum_k N_Q^k - \sqrt{\left(\sum_k N_Q^k\right)^2 + 2\sum_k N^Q * \overline{x^2} * \mu^2}}{-\sum_k N_Q^k + \sqrt{\left(\sum_k N_Q^k\right)^2 + 2\sum_k N^Q * \overline{x^2} * \mu^2}}$$

For exponential service, general service with Processor Sharing and deterministic service time, the above model gives the following dynamic models:

$$\dot{N}_Q^c(t) = \lambda^c(t) - \mu C(t) * N_Q^c(t) *$$

$$* \frac{2 + \sum_k N_Q^k - \sqrt{\left(\sum_k N_Q^k\right)^2 + 4 \sum_k N^Q}}{-\sum_k N_Q^k + \sqrt{\left(\sum_k N_Q^k\right)^2 + 4 \sum_k N^Q}} \quad M/M/1 \textit{ or } P.S.$$

$$\dot{N}_Q^c(t) = \lambda^c(t) - \mu C(t) * 2N_Q^c(t) *$$

$$* \frac{1 + \sum_k N_Q^k - \sqrt{\left(\sum_k N_Q^k\right)^2 + 2 \sum_k N^Q}}{-\sum_k N_Q^k + \sqrt{\left(\sum_k N_Q^k\right)^2 + 2 \sum_k N^Q}} \quad M/D/1$$

Note that for single class, we have:

$$\dot{N}_Q(t) = \lambda(t) - \mu C(t) * \frac{-N_Q(t) + \sqrt{(N_Q)^2 + 2N_Q * \overline{x^2} * \mu^2}}{\overline{x^2} * \mu^2} \quad M/G/1$$

$$\dot{N}_Q(t) = \lambda(t) - \mu C(t) * \frac{-N_Q(t) + \sqrt{(N_Q)^2 + 4N_Q}}{2} \quad M/M/1 \textit{ or } P.S.$$

$$\dot{N}_Q(t) = \lambda(t) - \mu C(t) * \left(-N_Q(t) + \sqrt{(N_Q)^2 + 2N_Q}\right) \quad M/D/1$$

5.4.4 Dynamic Queueing Models for Priority Classes

In this section, we derive dynamic models for queues with priority classes. For Poisson arrival and exponential service times. The average number of packets in the system of the high priority class α is given by

$$N^\alpha = \frac{\rho^\alpha}{1-\rho^\alpha}$$

and the average number of packets in the system of the low priority class β is given by

$$N^\beta = \frac{\rho^\beta * (1-\rho^\alpha) + \rho^\alpha \rho^\beta * \mu^\beta / \mu^\alpha}{(1-\rho^\alpha) * (1-\rho^\alpha-\rho^\beta)}$$

Solving the above system, we have the utilization of class α as a function of the average number of class α packets in the system and the utilization of class β as a function of the average number of class α and β packets in the system

$$\rho^\alpha = \frac{N^\alpha}{1+N^\alpha}$$

$$\rho^\beta = \frac{N^\beta}{(1+N^\alpha) * (1+N^\alpha * \dfrac{\mu^\beta}{\mu^\alpha} + N^\beta)}$$

Then, we have the following dynamic model for the high preemptive priority class α:

$$\rho^\alpha = \frac{N^\alpha(t)}{1+N^\alpha(t)}$$

and for the low preemptive priority class β:

$$\rho^\beta = \frac{N^\beta}{(1+N^\alpha(t)) * (1+N^\alpha(t) * \dfrac{\mu^\beta}{\mu^\alpha} + N^\beta)}$$

Next, we give a dynamic model for the average number of packets in the queue. The average number of packets in the queue of the high priority class α is given by

$$N_Q^\alpha = \frac{(\rho^\alpha)^2}{1-\rho^\alpha}$$

and the average number of packets in the queue of the low priority class β is given by

$$N_Q^\beta = \frac{\rho^\beta * (\rho^\alpha + \rho^\beta) * (1-\rho^\alpha) + \rho^\beta * \rho^\alpha * \mu^\beta/\mu^\alpha}{(1-\rho^\alpha) * (1-\rho^\alpha-\rho^\beta)}$$

Solving the above system, we have the utilization of class α as a function of the average number of class α packets in queue and the utilization of class β as a function of the average number of class α and β packets in queue

$$\rho^\alpha = \frac{-N_Q^\alpha + \sqrt{(N_Q^\alpha)^2 + 4N_Q^\alpha}}{2}$$

$$\rho^\beta = \left[-\frac{\rho^\alpha * \mu^\beta/\mu^\alpha + (1-\rho^\alpha) * (\rho^\alpha + N_Q^\beta)}{2(1-\rho^\alpha)} + \right.$$

$$\left. + \frac{\sqrt{[\rho^\alpha * \mu^\beta/\mu^\alpha + (1-\rho^\alpha) * (\rho^\alpha + N_Q^\beta)]^2 + 4N_Q^\beta(1-\rho^\alpha)^3}}{2(1-\rho^\alpha)}\right]$$

Then the dynamic model of the average number of packets in queue for the high preemptive priority class α is

$$\dot{N}_Q^\alpha(t) = \lambda^\alpha(t) - \mu^\alpha C(t) * \frac{-N_Q^\alpha(t) + \sqrt{(N_Q^\alpha(t))^2 + 4N_Q^\alpha(t)}}{2}$$

and for the low preemptive priority class β

$$\dot{N}_Q^\beta(t) = \lambda^\beta(t) - \mu^\beta C(t) * \left[-\frac{\rho^\alpha * \mu^\beta/\mu^\alpha + (1-\rho^\alpha) * (\rho^\alpha + N_Q^\beta(t))}{2(1-\rho^\alpha)} + \right.$$

$$\left. + \frac{\sqrt{[\rho^\alpha * \mu^\beta/\mu^\alpha + (1-\rho^\alpha) * (\rho^\alpha + N_Q^\beta(t))]^2 + 4N_Q^\beta(t)(1-\rho^\alpha)^3}}{2(1-\rho^\alpha)}\right]$$

Similarly, for Poisson arrival and exponential service times, we have the following dynamic model of the average number of packets for the high non-preemptive priority class α:

$$\dot{N}_Q^{\alpha}(t) = \lambda^{\alpha}(t) - \mu C(t) *$$

$$* \frac{N_Q^{\alpha} * (N_Q^{\alpha} + N_Q^{\beta}) + \sqrt{[N_Q^{\alpha} * (N_Q^{\alpha} + N_Q^{\beta})]^2 + 4(N_Q^{\alpha})^2 * (N_Q^{\alpha} + N_Q^{\beta} + N_Q^{\alpha} N_Q^{\beta})}}{2(N_Q^{\alpha} + N_Q^{\beta} + N_Q^{\alpha} N_Q^{\beta})}$$

and for the low non-preemptive priority class β

$$\dot{N}_Q^{\beta}(t) = \lambda^{\beta}(t) - \mu C(t) * \left(\frac{N_Q^{\alpha} * (1 - \rho^{\alpha})}{\rho^{\alpha}} - \rho^{\alpha} \right)$$

5.4.5 Second Order Dynamic Queueing Models

In this section, we suggest using a second order model in optimization of systems with bursty traffic, where the variance of the number of packets can be large. Next, we suggest using the second order model by Rothkopf & Oren [407], and Clark [109]:

$$\dot{N}(t) = \lambda(t) - \mu C(t) * (1 - \pi_0(t))$$

$$Var(N(t)) = \lambda(t) + \mu C(t) - \mu C(t) * \pi_0(t) * (2N(t) + 1)$$

where

$$\pi_0(t) = \left(\frac{N(t)}{Var(N(t))} \right)^{\frac{(N(t))^2}{Var(N(t)) - N(t)}}$$

5.4.6 Wiener Process Models

In this section, we introduce a Wiener process model for flow that fluctuates with a large variance around its average value. We introduce a stochastic term for the arrival and departure rate in the dynamic models presented in the previous section. For example, the dynamic model for $M/M/1$ queues becomes:

$$\dot{N}^c(t) = \left(\lambda^c(t) - a^c(t) * \frac{dw_a^c}{dt} \right) - \left(\mu C(t) * \frac{N^c(t)}{1 + \sum_k N^k(t)} - b^c(t) * \frac{dw_b^c}{dt} \right)$$

where $a^c(t)$ and $b^c(t)$ are the standard deviations of the arrival and departure rates for class c, and $w_a^c(t)$ and $w_b^c(t)$ are Wiener processes.

We can rewrite the above model as

$$\dot{N}^c(t) = \left(\lambda^c(t) - a^c(t) * \overline{\xi^c}_a(t)\right) - \left(\mu C(t) * \frac{N^c(t)}{1 + \sum_k N^k(t)} - b^c(t) * \overline{\xi^c}_b(t)\right)$$

where $\overline{\xi^c}_a(t) = \frac{dw_a^c(t)}{dt}$ and $\overline{\xi^c}_b(t) = \frac{dw_b^c(t)}{dt}$ are zero mean, unit variance normal random variables.

5.4.7 Cost Functions

In this section, we introduce cost functions that can be used in the dynamic problem. Desired properties of a cost function are to be: i) nonnegative, ii) nondecreasing, iii) continuous and iv) convex.

We may consider as cost function the total time packets spent on each network resource ij

$$g^c_{ij[sd]} = \int_{t_0}^{t_f} N^c_{ij[sd]}(t)dt$$

blocking at resource ij

$$g^c_{ij[sd]} = \int_{t_0}^{t_f} B^c_{ij[sd]}(t)dt$$

blocking at path $\pi[sd]$

$$g^c_{\pi[sd]} = \int_{t_0}^{t_f} \prod_{ij \in \pi[sd]} B^c_{ij[sd]}(t)dt$$

rejected flow at resource ij

$$g^c_{ij[sd]} = \int_{t_0}^{t_f} \phi^c_{o,ij[sd]}(t)dt$$

rejected flow at source $[s.]$ for destination $[.d]$

$$g^c_{0,[sd]} = \int_{t_0}^{t_f} \phi^c_{o,[sd]}(t)dt$$

5.4.8 State Constraints

In this section, we define flow control constraints on the number of packets $N_{ij[sd]}(t) \geq 0$ that can coexist at the network resources. For clear exposition, we consider only one class in the network. The case of multiple classes follows trivially.

The total expected number of packets on every link ij should be less than the buffer (or window) size of link ij, $\sum\limits_{[sd]} N_{ij[sd]}(t) \leq W_{ij}(t)$.

Also, in order to guarantee an upper bound on the delay that packets may suffer from source to destination, the total expected number of packets on every path $\pi[sd]$ should be less than the end-to-end window size on path $\pi[sd]$, $\sum\limits_{[s_1 d_1]} \sum\limits_{ij} N_{ij[s_1 d_1]}(t) * 1_{ij \in \pi[sd]}(t) \leq W_{\pi[sd]}$, where $1_{ij \in \pi[sd]}(t)$ is the indicator function that link ij is on the path $\pi[sd]$.

Although controlling the total number of packets in the network is optimum from the system point of view, it may also be unfair to some users. Some aggressive users (a source, a destination or a virtual circuit) may congest the network. If the flow and congestion control operate without paying attention to the identity of packets, other users may be unfairly penalized. So, we point out three identities that should also be controlled for fairness reasons:

1) *each source*:

In order that source $[s.]$ does not monopolize the network resources, the total expected number of packets originated from node $[s.]$ should be less than the network "capacity" for packets from source $[s.]$, $\sum\limits_{[.d]} \sum\limits_{ij} N_{ij[sd]}(t) \leq W_{[s.]}$. Also, in order that source $[s.]$ does not monopolize path $\pi[sd]$, the total expected number of packets originated at node $[s.]$ on path $\pi[sd]$ should be less than the path's "capacity" for packets originated at $[s.]$, $\sum_{[.d_1]}\sum_{ij} N_{ij[sd_1]}(t) * 1_{ij\in\pi[sd]}(t) \leq W_{[s.],\pi[sd]}$. Finally, in order that source $[s.]$ does not monopolize link ij, the total expected number of packets originated at node $[s.]$ on link ij should be less than the link's "capacity" for packets originated at $[s.]$, $\sum_{[.d]} N_{ij[sd]}(t) \leq W_{[s.],ij}$.

2) *each destination*:

Similarly, in order that destination $[.d]$ does not monopolize the network resources, the total expected number of packets destined to node $[.d]$ should be less than the network "capacity" for packets to destination $[.d]$, $\sum_{[s.]}\sum_{ij} N_{ij[sd]}(t) \leq W_{[.d]}$. Also, in order that destination $[.d]$ does not monopolize path $\pi[sd]$, the total expected number of packets destined to node $[.d]$ on path $\pi[sd]$ should be less than the path's "capacity" for packets destined to $[.d]$, $\sum_{[s_1.]}\sum_{ij} N_{ij[s_1d]}(t) * 1_{ij\in\pi[sd]}(t) \leq W_{[.d],\pi[sd]}$. Finally, in order that destination $[.d]$ does not monopolize link ij, the total expected number of packets destined to node $[.d]$ on link ij should be less than the link's "capacity" for packets destined to $[.d]$, $\sum_{[s.]} N_{ij[sd]}(t) \leq W_{[.d],ij}$.

3) *each class*:

In order that $[sd]$ packets do not monopolize the network resources, the total expected number of $[sd]$ packets should be less than the network capacity for $[sd]$ packets, $\sum_{ij} N_{ij[sd]}(t) \leq W_{[sd]}$. Also, in order that $[sd]$ packets do not monopolize path $\pi[sd]$, the total expected number of $[sd]$ packets on path $\pi[sd]$ should be less than the end-to-end window size for $[sd]$ packets on path $\pi[sd]$, $\sum_{ij} N_{ij[sd]}(t) * 1_{ij\in\pi[sd]}(t) \leq W_{\pi[sd],[sd]}$. Finally, in order that $[sd]$ packets do not monopolize link ij, the expected number of $[sd]$ packets on link ij, should be less than the buffer (or window) size for $[sd]$ packets on link ij, $N_{ij[sd]}(t) \leq W_{ij[sd]}(t)$, with $\sum_{[sd]} W_{ij[sd]}(t) \geq W_{ij}(t)$.

5.5 Application to Virtual Circuit Networks

In this section, we apply the methodologies developed in the previous sections to virtual circuit networks. We develop dynamic queueing models for the average number of packets coupled with dynamic queueing models for the average number of virtual circuits. Furthermore, we introduce Wiener process models for modeling the stochastic system. We also present some cost functions and state constraints that can be used in the optimal control problem. Finally, we propose a class of heuristic link lengths that can be used on-line.

5.5.1 Dynamic Queueing Models for Multiple Classes

We consider two levels of the flow in virtual circuit networks. At the virtual level, we model each network resource as an $M/M/\infty$ queue. That means that an infinite number of virtual circuits may coexist at every network resource (see section 5.6.2). Then the average number of class c virtual circuits is described by the following dynamic model:

$$\dot{V}^c(t) = \gamma^c(t) - \delta^c(t) * V^c(t)$$

At the packet level, we model each network resource using any model among those proposed in section 5.4 for datagram networks. For example, the average number of class c packets for $M/M/1$ or $P.S.$ queues is described by the following dynamic model:

$$\dot{N}^c(t) = r^c(t) * V^c(t) - \mu C(t) * \frac{N^c(t)}{1 + \sum_k N^k}$$

where $r^c(t)$ is the average packet arrival rate per virtual circuit of class c.

Next, we give several dynamic models for virtual circuit networks for different optimization formulations. For easy of exposition, we consider dynamic models only for links.

i) The routing decisions are done at each source node [s.] on the path flow space. Therefore, the total arrival rate to link ij is the sum of all path flows that pass through this link:

$$\dot{V}_{ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) - \delta_{[sd]}(t) * V_{ij[sd]}(t)$$

ii) The routing decisions are done at each network node on the link flow space and the virtual circuit duration is very small. Therefore the virtual circuit departure rate from a link becomes arrival rate to the next link. The arrival rate to the outgoing links sj from the source node $[s.]$ is the fraction of flow that is routed through that link. The arrival rate to an intermediate link ij is the departure rate from the ingoing links to node i weighted by the fraction $\phi_{ij[sd]}$ that is assigned to outgoing link ij from node i:

$$\dot{V}_{sj[sd]}(t) = \gamma_{[sd]}(t) * \phi_{sj[sd]}(t) - \delta_{[sd]}(t) * V_{sj[sd]}(t)$$
$$\dot{V}_{ij[sd]}(t) = \sum_{k\in I(i)} \delta_{[sd]}(t) * V_{ki[sd]}(t) * \phi_{ij[sd]}(t) - \delta_{[sd]}(t) * V_{ij[sd]}(t) \quad s \neq i$$

iii) The routing decisions are done at each network node on the path flow space and the virtual circuit duration is very small. Therefore the virtual circuit departure rate from a link becomes arrival rate to the next link. The arrival rate to the outgoing links sj from the source node $[s.]$ is the sum of all fractions of path flows that are routed through that link. The arrival rate to an intermediate link ij is the departure rate from the ingoing links to node i that is assigned to outgoing link ij from node i:

$$\dot{V}_{sj[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{sj\in\pi[sd]}(t) - \delta_{[sd]}(t) * V_{sj[sd]}(t)$$
$$\dot{V}_{ij[sd]}(t) = \sum_{k\in I(i)} \sum_{\pi[sd]} \delta_{[sd]}(t) * V_{ki[sd]}(t) * 1_{ki\in\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t)$$
$$-\delta_{[sd]}(t) * V_{ij[sd]}(t) \quad s \neq i$$

iv) A dynamic model for the paths (virtual and not physical links): The average number of virtual circuits for $[sd]$ on path $\pi[sd]$ is:

$$\dot{V}_{\pi[sd]}(t) = \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) - \delta_{[sd]}(t) * V_{\pi[sd]}(t)$$

v) The routing decisions are done at each network node on the path flow space and the virtual circuit duration is very long. Therefore a virtual circuit stays at each network link for long time, so we can assume that its arrival rate does not change drastically over time. Then the arrival rate at each link is the sum of path flows that pass through this link:

$$\dot{V}_{ij}(t) = \sum_{[sd]} \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij \in \pi[sd]}(t) - \delta(t) * V_{ij}(t)$$

vi) The routing decisions are done at each network node on the link flow space. However, if a virtual circuit is rejected with probability $\phi_{no[sd]}$ (for congestion control reasons) at node n, then this virtual circuit is reestablished from the source. Therefore, the successful arrival rate at the source $[s.]$ is $\gamma_{[sd]}(t) * \prod_n (1 - \phi_{no[sd]}(t))$ Virtual circuits may also be rejected at each network link for congestion control reasons:

$$\dot{V}_{sj[sd]}(t) = \gamma_{[sd]}(t) * \prod_n (1 - \phi_{no[sd]}(t)) * \phi_{sj[sd]}(t) - \delta_{[sd]}(t) * V_{sj[sd]}(t)$$
$$\dot{V}_{ij[sd]}(t) = \sum_{k \in I(i)} \delta_{[sd]}(t) * V_{ki[sd]}(t) * \phi_{ij[sd]}(t) - \delta_{[sd]}(t) * V_{ij[sd]}(t) \quad s \neq i$$

Next, we give several dynamic models for the packet level for different optimization formulations:

i) The routing decisions are done at each network node on the link flow space and the virtual circuit duration is very long. Then a virtual circuit stays for long time at each network link:

$$\dot{N}_{ij[sd]}(t) = r_{[sd]}(t) * V_{ij[sd]}(t) - \mu C_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t))$$

ii) The routing decisions are done at each network node on the link flow space. The packet departure rate from a node is routed to an outgoing link from that node:

$$\dot{N}_{sj[sd]}(t) = r_{[sd]}(t) * V_{sj[sd]}(t) - \mu C_{sj}(t) * \rho_{sj[sd]}(\mathbf{N}_{sj}(t))$$

$$\dot{N}_{ij[sd]}(t) = \sum_{k \in I(i)} \mu C_{ki}(t) * \rho_{ki[sd]}(\mathbf{N}_{ki}(t)) * \phi_{ij[sd]}(t)$$

$$-\mu C_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t)) \quad s \neq i$$

iii) The routing decisions are done at each network node on the link flow space. The packet departure rate from a node is routed to an outgoing link from that node. However, packets may also fail transmission on link ki, that has error rate $e_{ki}(t)$ and be retransmitted from the source node $[s.]$ after a time-out period τ. So, the source node $[s.$ receives an extra $[sd]$ flow, $\sum_{ki} \mu C_{ki}(t-\tau) * e_{ki}(t-\tau) * \rho_{ki[sd]}(\mathbf{N}_{ki}(t-\tau))$, due to packet failures at links ki inside the network. Then the source node $[s.]$ routes a fraction $\phi_{sj[sd]}(t)$ of this flow to its outgoing link sj. Any other node $i \neq s$, receives the successful packet flow, from its input neighbors and routes a fraction $\phi_{ij[sd]}(t)$ of this flow to its outgoing link ij.

$$\begin{aligned}
\dot{N}_{sj[sd]}(t) = \; & r_{[sd]}(t) * V_{sj[sd]}(t) + \\
& + \sum_{ki} \mu C_{ki}(t-\tau) * e_{ki}(t-\tau) * \rho_{ki[sd]}(\mathbf{N}_{ki}(t-\tau)) * \phi_{sj[sd]} \\
& - \mu C_{sj}(t) * \rho_{sj[sd]}(\mathbf{N}_{sj}(t)) \\
\dot{N}_{ij[sd]}(t) = \; & \sum_{k \in I(i)} \mu C_{ki}(t) * (1 - e_{ki}(t)) * \rho_{ki[sd]}(\mathbf{N}_{ki}(t)) * \phi_{ij[sd]}(t) \\
& - \mu C_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t)) \quad s \neq i
\end{aligned}$$

iv) The routing decisions are done at each network node on the link flow space and the virtual circuit duration is long. The packet departure rate from a node is routed to an outgoing link from that node. However, packets may also fail at a link ki, that has error rate $e_{ki}(t)$ and be retransmitted from the source after time τ:

$$\dot{N}_{sj[sd]}(t) = r_{[sd]}(t) * V_{sj[sd]}(t) + \sum_{ki} r_{[sd]}(t) * V_{ki[sd]}(t) * e_{ki(t)}$$

$$-\mu C_{sj}(t) * \rho_{sj[sd]}(\mathbf{N}_{sj}(t))$$

$$\dot{N}_{ij[sd]}(t) = \sum_{k \in I(i)} r_{[sd]}(t) * V_{ki[sd]}(t) * (1 - e_{ki}(t)) * \phi_{ij[sd]}(t)$$

$$-\mu C_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t)) \quad s \neq i$$

v) The routing decisions are done at each network node on the path flow space and the virtual circuit duration is long.

$$\dot{N}_{ij[sd]}(t) = \sum_{\pi[sd]} r_{[sd]}(t) * V_{\pi[sd]}(t) * 1_{ij \in \pi[sd]} - \mu C_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t))$$

vi) Finally, if we do not consider classes

$$\dot{N}_{ij}(t) = r(t) * V_{ij}(t) - \mu C_{ij}(t) * \rho_{ij}(\mathbf{N}_{ij}(t))$$

5.5.2 Cost Functions

In this section, we introduce cost functions for the dynamic problem. We consider as cost function for class c and source-destination $[sd]$, at time t, at network resource ij, the total time packets spent at ij

$$g^c_{ij[sd]}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) = C^c_{N,i[sd]} * N^c_{i[sd]}(t)$$

the total time virtual circuits spent at ij,

$$g^c_{ij[sd]}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) = C^c_{V,i[sd]} * V^c_{i[sd]}(t)$$

the rejected flow cost at ij

$$g^c_{ij[sd]}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) = C^c_{o[sd]} * \lambda^c_{o[sd]}(t)$$

the negative throughput at ij

$$g^c_{ij[sd]}(t, \mathbf{X}(t), \mathbf{\Phi}(t), \mathbf{\Psi}(t)) = C'^c_{\mu,i[sd]} * \mu C_{ij} * \rho^c_{ij[sd]}(\mathbf{N}_{ij}(t))$$

5.5.3 Length of a Link

In previous sections, we found that jobs are sent to destinations of minimum length and are routed through minimum length paths. At that time we defined the lengths to destinations, the path lengths and the rejection lengths. In this section, we introduce some lengths that are very simple, however they are based on heuristic arguments.

For dynamic (or adaptive) load sharing, routing and congestion control, we need to know the state of each system resource. We define as length of a system resource the load on this resource. So, if a resource is lightly loaded, then its length is small, while if the resource is heavily loaded, then its length is large. Then the dynamic algorithm chooses the resource with the minimum length. Depending on the information we select about the state of each system resource, we may define different lengths of the resource (link, node, computer site, etc.).

Next, we define the length of a resource at time t as a convex combination of its current length at t and its expected length in the future:

$$l_{ij}(t) = \epsilon_{ij}(t) * l_{ij}^{current}(t) + (1 - \epsilon_{ij}(t)) * l_{ij}^{future}(t) \quad 0 \leq \epsilon \leq 1$$

Based on models presented earlier, we define some simple approximations for these lengths:

$$
\begin{aligned}
l_{ij}^{current}(t) &= \frac{1 + N_{ij}(t)}{\mu C_{ij}(t)} \\
&= T_{ij}(t) \\
&= \begin{cases} \dfrac{1}{\mu C_{ij}(t) - r(t) * V_{ij}(t)} & \text{if } \mu C_{ij}(t) > r(t) * V_{ij}(t) \\ \infty & \text{o.w.} \end{cases} \\
&= \begin{cases} \dfrac{1}{\mu C_{ij}(t) - \lambda_{ij}(t)} & \text{if } \mu C_{ij}(t) > \lambda_{ij}(t) \\ \infty & \text{o.w.} \end{cases} \\
&= \frac{[1 + N_{ij}(t)]^2}{\mu C_{ij}(t)} \\
&= \begin{cases} \dfrac{\mu C_{ij}(t)}{[\mu C_{ij}(t) - r(t) * V_{ij}(t)]^2} & \text{if } \mu C(t)_{ij} > r * V_{ij}(t) \\ \infty & \text{o.w.} \end{cases} \\
&= \begin{cases} \dfrac{\mu C_{ij}(t)}{[\mu C_{ij}(t) - \lambda_{ij}(t)]^2} & \text{if } \mu C_{ij}(t) > \lambda_{ij}(t) \\ \infty & \text{o.w.} \end{cases} \\
&= \begin{cases} \dfrac{\mu C_{ij}(t)}{[\mu C_{ij}(t) - \lambda_{ij}(t)] * [\mu C_{ij}(t) - \lambda_{ij}(t) - r(t)]} \\ \qquad\qquad \text{if } \mu C_{ij}(t) > \lambda_{ij}(t) - r(t) \\ \infty \qquad \text{o.w.} \end{cases} \\
&= \frac{1}{\mu C_{ij}(t) * [1 - \rho_{ij}(t)]^2}
\end{aligned}
$$

$$l_{ij}^{future} = \frac{1+V_{ij}(t)}{\mu C_{ij}(t)}$$

$$= \begin{cases} \frac{1}{\mu C_{ij}(t) - r(t) * [1+V_{ij}(t)]} & \text{if } \mu C_{ij}(t) > r(t) * [1+V_{ij}(t)] \\ \infty & \text{o.w.} \end{cases}$$

$$= \begin{cases} \frac{1}{\mu C_{ij}(t) - \lambda_{ij}(t) - r(t)} & \text{if } \mu C_{ij}(t) > \lambda_{ij} + r(t) \\ \infty & \text{o.w.} \end{cases}$$

5.5.4 State Constraints

In this section, we define constraints on the number of virtual circuits $V_{ij[sd]}(t) \geq 0$ and packets $N_{ij[sd]}(t) \geq 0$ at the network resources.

The total expected number of virtual circuits at every link ij should be less than the virtual circuit "capacity" on link ij (for example the number of buffers), $\sum_{[sd]} V_{ij[sd]}(t) \leq \Omega_{ij}(t)$.

Although controlling the total number of virtual circuits and packets in the network is optimum from the system point of view, it may also be unfair to some users. Some aggressive users (a source, a destination or a virtual circuit) may congest the network. If the flow and congestion control operate without paying attention to the identity of virtual circuits and packets, other users may be unfairly penalized. So, we point out three identities that should also be controlled for fairness reasons:

1) *each source*:

In order that source $[s.]$ does not monopolize the network resources, the total expected number of virtual circuits originated from node $[s.]$ should be less than the network "capacity" for virtual circuits from source $[s.]$, $\sum_{[.d]} \sum_{ij} V_{ij[sd]}(t) \leq \Omega_{[s.]}$. Also, in order that source $[s.]$ does not monopolize path $\pi[sd]$, the total expected number of virtual circuits originated at node $[s.]$ on path $\pi[sd]$ should be less than the path's "capacity" for virtual circuits originated at $[s.]$, $\sum_{[.d1]} \sum_{ij} V_{ij[sd_1]}(t) *$

$1_{ij\in\pi[sd]}(t) \leq \Omega_{[s.],\pi[sd]}$. Finally, in order that source $[s.]$ does not monopolize link ij, the total expected number of virtual circuits originated at node $[s.]$ on link ij should be less than the link's "capacity" for virtual circuits originated at $[s.]$, $\sum_{[.d]} V_{ij[sd]}(t) \leq \Omega_{[s.],ij}$.

Similarly, we may impose restrictions on the number of packets as for datagram networks (see section 5.4.8).

2) *each destination*:

Similarly, in order that destination $[.d]$ does not monopolize the network resources, the total expected number of virtual circuits destined to node $[.d]$ should be less than the network "capacity" for virtual circuits to destination $[.d]$, $\sum_{[s.]}\sum_{ij} V_{ij[sd]}(t) \leq \Omega_{[.d]}$. Also, in order that destination $[.d]$ does not monopolize path $\pi[sd]$, the total expected number of virtual circuits destined to node $[.d]$ on path $\pi[sd]$ should be less than the path's "capacity" for virtual circuits destined to $[.d]$, $\sum_{[s_1.]}\sum_{ij} V_{ij[s_1d]}(t) * 1_{ij\in\pi[sd]}(t) \leq \Omega_{[.d],\pi[sd]}$. Finally, in order that destination $[.d]$ does not monopolize link ij, the total expected number of virtual circuits destined to node $[.d]$ on link ij should be less than the link's "capacity" for virtual circuits destined to $[.d]$, $\sum_{[s.]} V_{ij[sd]}(t) \leq \Omega_{[.d],ij}$.

Similarly, we may impose restrictions on the number of packets as for datagram networks (see section 5.4.8).

3) *each class or each single virtual circuit*:

We can consider each virtual circuit as a different class, therefore the following restrictions which apply for each $[sd]$ class of virtual circuits may also apply for each virtual circuit. So, in order that $[sd]$ virtual circuits do not monopolize the network resources, the total expected number of $[sd]$ virtual circuits should be less than the network capacity for $[sd]$ virtual circuits, $\sum_{ij} V_{ij[sd]}(t) \leq \Omega_{[sd]}$. Also, in order that $[sd]$ virtual circuits do not monopolize path $\pi[sd]$, the total expected number of $[sd]$ virtual circuits on path $\pi[sd]$ should be less than the end-to-end window size for $[sd]$ virtual circuits on path $\pi[sd]$, $\sum_{ij} V_{ij[sd]}(t) * 1_{ij\in\pi[sd]}(t) \leq \Omega_{\pi[sd],[sd]}$. Finally, in order that $[sd]$ virtual circuits do not monopolize link ij, the expected number of $[sd]$ virtual circuits link ij, should be less than the buffer (or window) size for $[sd]$ virtual circuits on link ij, $V_{ij[sd]}(t) \leq \Omega_{ij[sd]}(t)$.

Similarly, we may impose restrictions on the number of packets as for datagram networks (see section 5.4.8).

5.6 Example

5.6.1 Introduction

In this section, we solve the decentralized dynamic joint routing and congestion control problem for multi-class multi-destination dynamic virtual circuit networks.

Two of the most important algorithms for efficient virtual circuit network control are routing and congestion control. *Routing* decides which route the virtual circuit will follow from source to destination. *Congestion control* prevents network overload by controlling the virtual circuit traffic entering the network. Routing and congestion control are strongly related problems and each affects the other. For a more accurate model and better network performance, both problems should be modeled and solved simultaneously. Such an approach however may increase the modeling and optimization complexity. Previous studies on virtual circuit network control usually concentrate on the routing problem.

In virtual circuit networks, a call set-up packet, which may be part of the first packet of a message, initiates the establishment of a virtual path from source to destination. All other packets belonging to this message follow the same route which remains fixed for the duration of the call. In this way, a virtual circuit provides a reliable logical channel with packets delivered in order. The route selection for each virtual circuit is the virtual circuit routing problem.

First, we introduce a nonlinear dynamic queueing model for virtual circuit networks that considers the dynamic interaction among the multi-class multi-destination virtual circuit and packet processes. We also define a multi-objective cost function of rejecting, setting up & maintaining virtual circuits, as well as of the packet delay and throughput.

Then we formulate the joint problem as an optimal control problem. Necessary optimality conditions are provided by Pontryagin's maximum principle. Sufficient optimality conditions based on the convexity of the Hamiltonian function are also given. For the finite horizon, the optimal controls can be found after numerically solving a Two-Point Boundary-Value Problem. For the long-run stationary equilibrium, we derive the state dependent routing and congestion controls. We show (via simulation) that when the updating period is not much larger than the mean interarrival time of virtual circuits, this state dependent routing algorithm and a shortest queue routing algorithm are showed (via simulation) to be superior to the optimal quasi-static routing.

5.6.2 Virtual Circuit Network Model

Consider an arbitrary network topology with multiple classes of virtual circuit traffic between multiple source-destination pairs (Figure 5.1)

Instead of introducing an extra notational index for each class of virtual circuits, we can consider each class c of virtual circuits between a source-destination pair $[sd]$ as being established between a fictitious $[s_c d_c]$ pair, where physically $s_c = s$ and $d_c = d, \ \forall c$. The queueing models that we introduce in this section can handle this substitution. Note also that one extreme case is to consider each virtual circuit as a different class. Another extreme case is to consider all virtual circuits as belonging to the same class. Also, in contemporary networks, the nodal processing delays are negligible compared to the transmission and propagation delays and therefore they were ignored in network optimization and control procedures. However, in future high speed networks, the transmission delays will be very short and comparable to the nodal processing delays. Therefore, packets will be queued not only in front of the links but also in front of the nodes (Figure 5.2). However, instead of introducing extra variables to describe the state of each node, we can consider each node i as a link $i_1 i_2$. So, in the following analysis, the word "link" may mean physically either a link or a node.

Virtual circuits arrive at a source node s (according to a Poisson distribution) destined for a destination node d with rate $\gamma_{[sd]}(t) \geq 0$ (Figure 5.3).

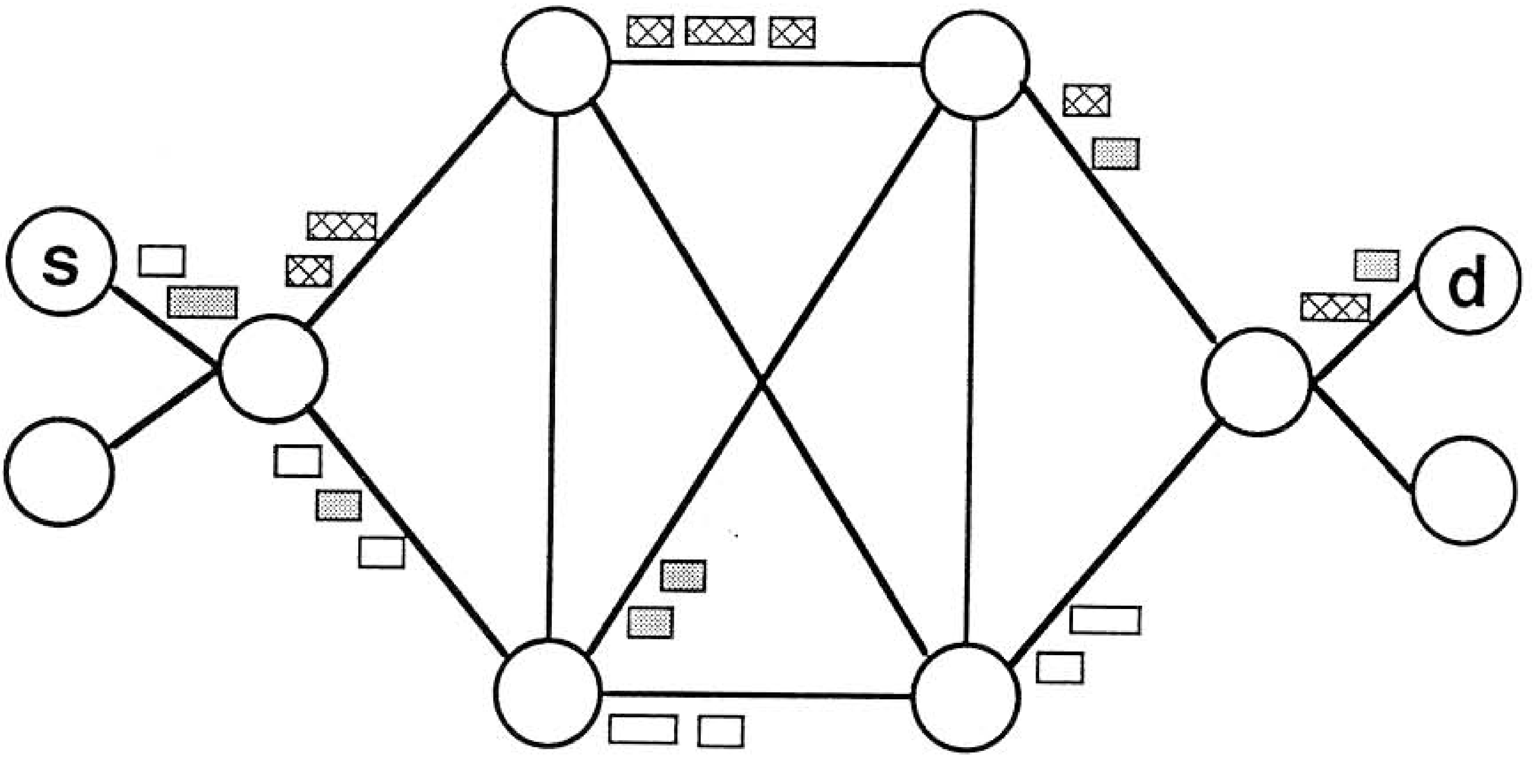

Figure 5.1: A Virtual Circuit Network.

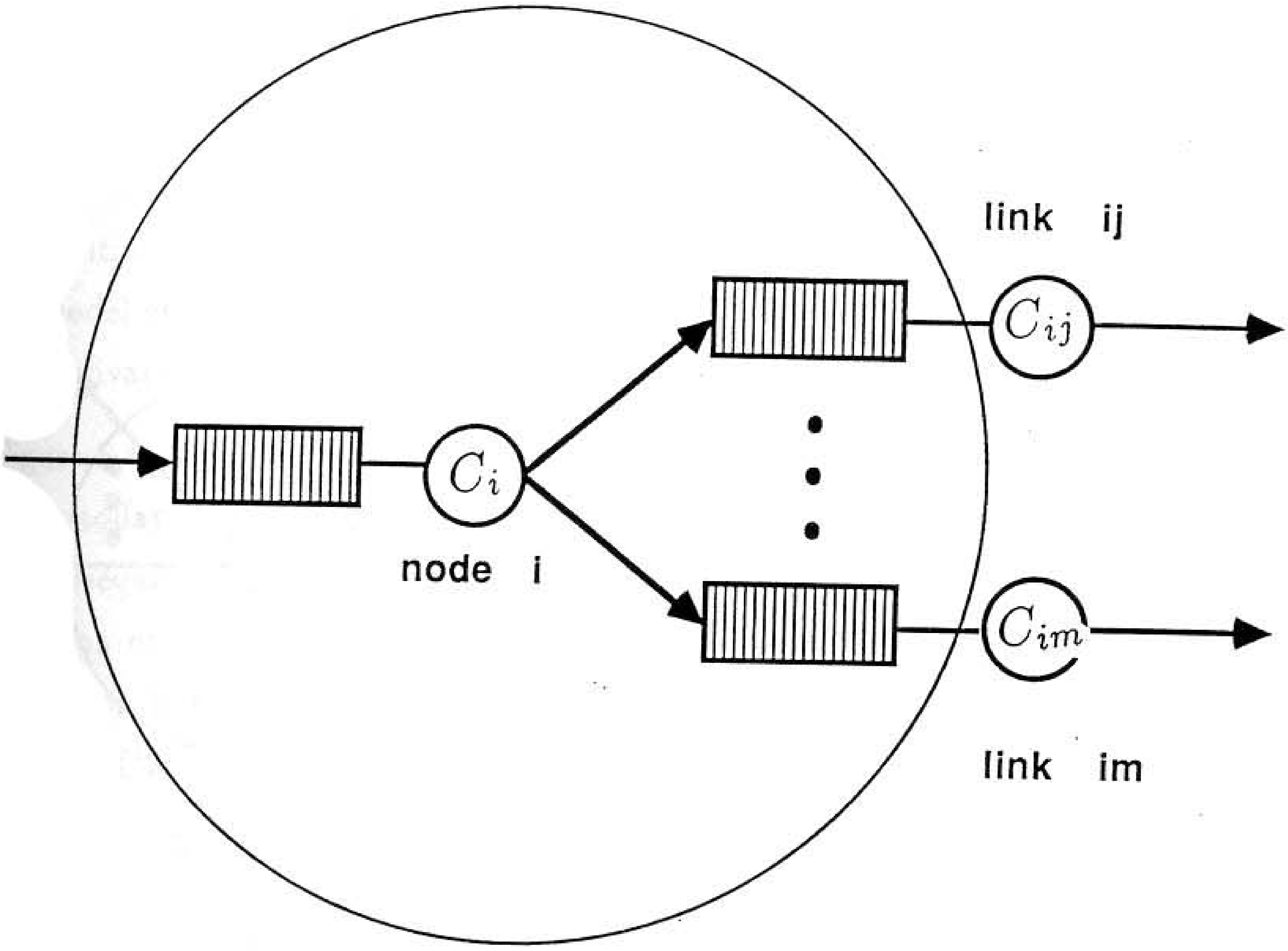

Figure 5.2: A network node.

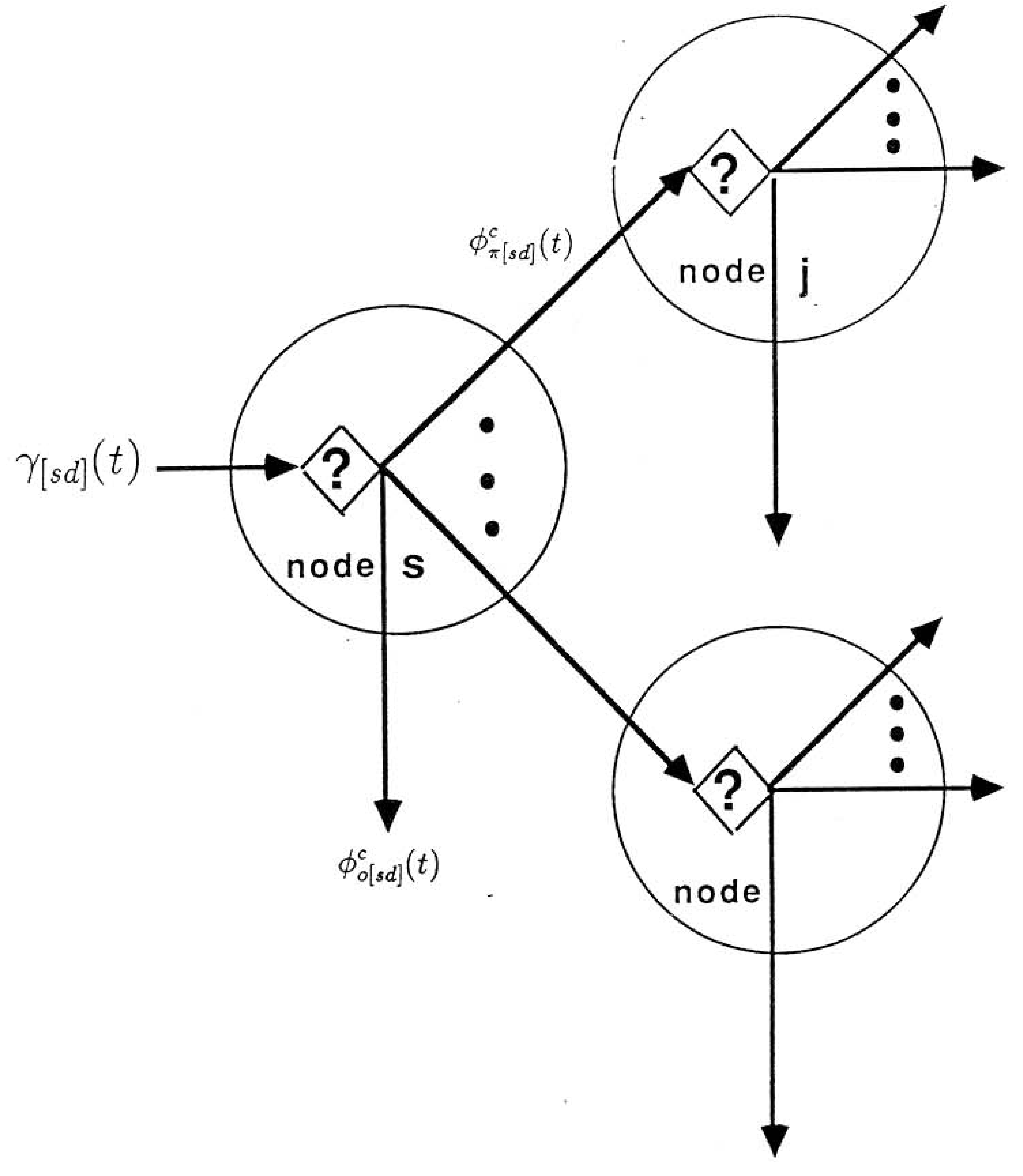

Figure 5.3: Virtual circuit routing and congestion control

For congestion control reasons, a fraction $\phi_{o[sd]}(t) \in [0,1]$ of these externally arriving $[sd]$ virtual circuits is rejected, while the remaining virtual circuits are accepted into the network. A fraction $\phi_{\pi[sd]}(t) \in [0,1]$ of the externally arriving $[sd]$ virtual circuits are routed from node s to its destination node d through path $\pi[sd]$, where $\phi_{o[sd]}(t) + \sum_{\pi[sd]} \phi_{\pi[sd]}(t) = 1$. Then the rejected $[sd]$ virtual circuit flow at the source node s is $\gamma_{[sd]}(t) * \phi_{o[sd]}(t)$ and the $[sd]$ virtual circuit flow on path $\pi[sd]$ is $\gamma_{[sd]}(t) * \phi_{\pi[sd]}(t)$. The above procedure happens for every source-destination pair in the network. Therefore the $[sd]$ virtual circuit flow on link ij is the sum of the $[sd]$ virtual circuit flows of all paths traversing this link, i.e. $\sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t)$.

Finally, each $[sd]$ virtual circuit stays in the network for some time duration exponentially distributed with mean $1/\delta_{[sd]}(t) \geq 0$ and then terminates. So, we can model every link ij for the $[sd]$ virtual circuit process as an $M/M/\infty$ queue with arrival rate $\sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t)$ and mean service time $1/\delta_{[sd]}(t)$ (Figure 5.4) We note that thousands of virtual circuits can coexist on a link (well within today's technology capabilities) [247].

Subsequently, we will introduce a state space approach to model the dynamic evolution of the virtual circuit processes. The expected number of $[sd]$ virtual circuits on link ij at time t, $V_{ij[sd]}(t) \geq 0$, increases during Δt by the expected number of $[sd]$ virtual circuits that arrive during this period, $\sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) * \Delta t$, minus the expected number of $[sd]$ virtual circuits that depart during this period, $\delta_{[sd]}(t) * V_{ij[sd]}(t) * \Delta t$ (Figure 5.5). So, the $[sd]$ virtual circuit process at link ij is described by

$$V_{ij[sd]}(t+\Delta t) = V_{ij[sd]}(t) + \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) * \Delta t$$
$$-\delta_{[sd]}(t) * V_{ij[sd]}(t) * \Delta t \quad \forall\, ij \quad \forall\, [sd]$$

The expected number of $[sd]$ virtual circuits on every link ij at time t, $V_{ij[sd]}(t)$, is a continuous function of time, so let us define

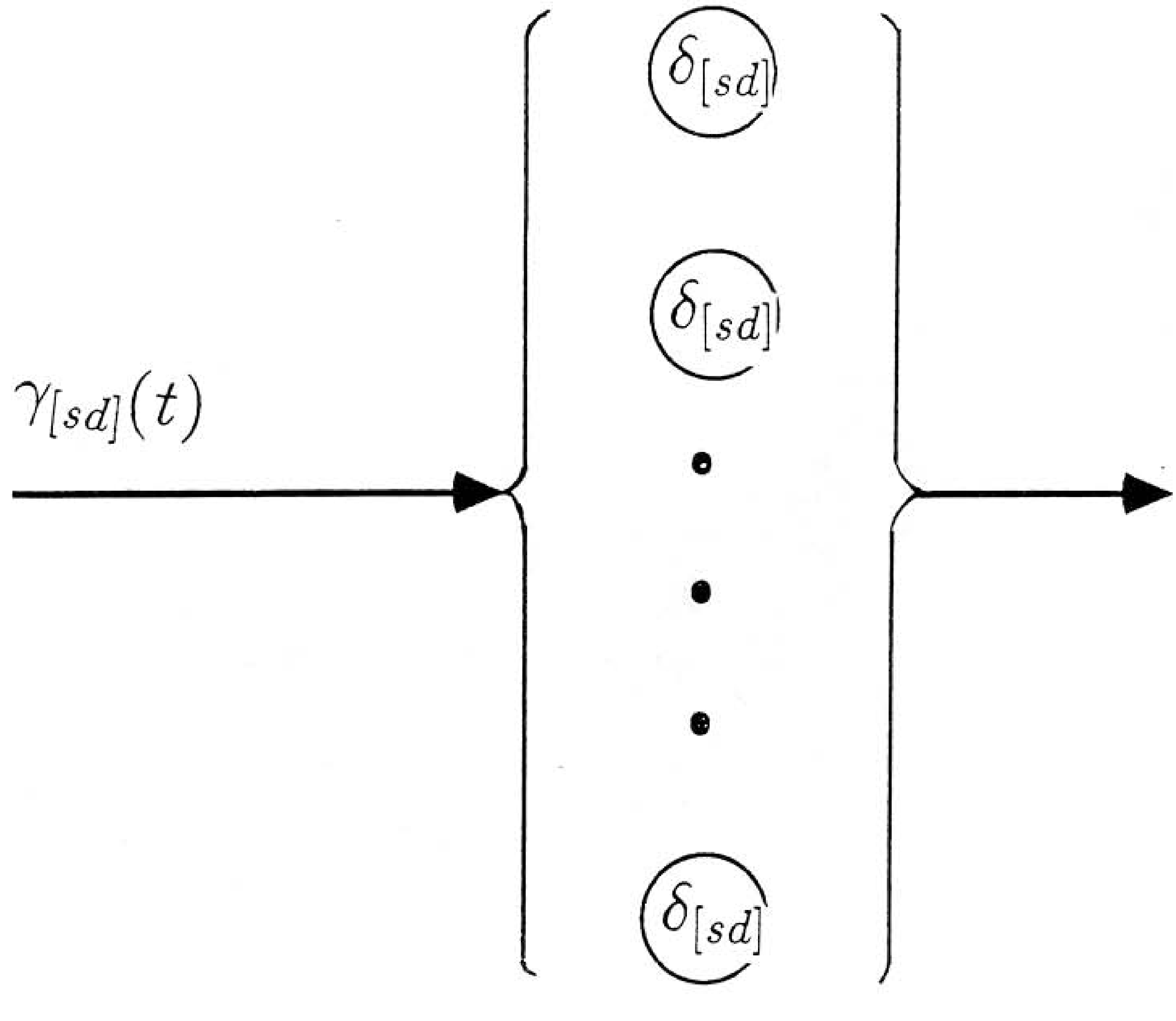

Figure 5.4: $M/M/\infty$ model for virtual circuit process.

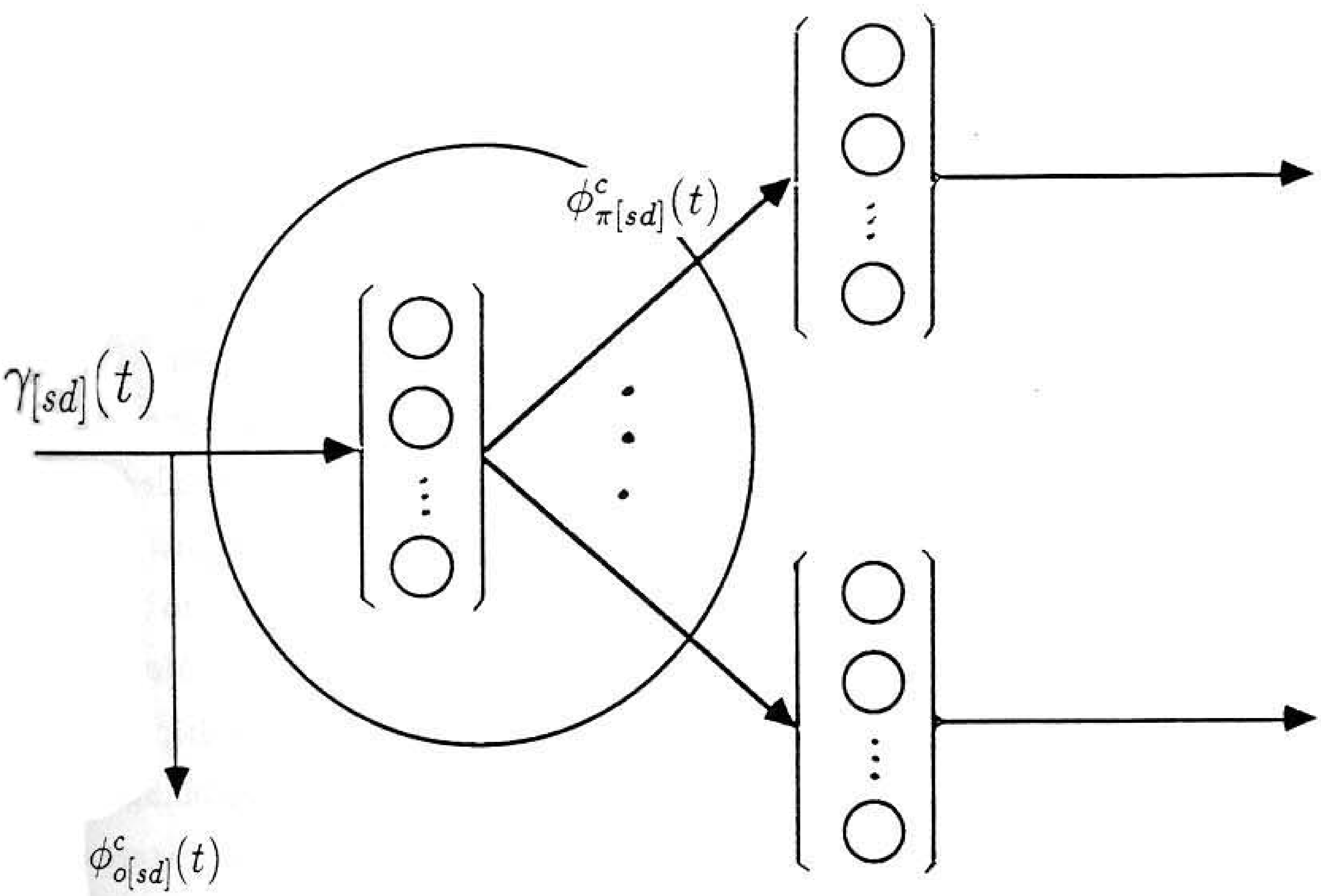

Figure 5.5: Virtual circuit processes.

$$\dot{V}_{ij[sd]}(t) = \lim_{\Delta t \to 0} \frac{V_{ij[sd]}(t+\Delta t) - V_{ij[sd]}(t)}{\Delta t} \qquad \forall\ ij \quad \forall\ [sd]$$

Therefore the $[sd]$ virtual circuit process on link ij at time t is described by

$$\dot{V}_{ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) - \delta_{[sd]}(t) * V_{ij[sd]}(t) \qquad \forall\ ij \quad \forall\ [sd]$$

Next, we describe the evolution of the packet process into the network. Let $r_{[sd]}(t) \geq 0$ be the packet arrival rate per $[sd]$ virtual circuit at time t (Poisson distribution) (Figure 5.6) If there are $V_{ij[sd]}(t)$ $[sd]$ virtual circuits on link ij at time t, then the total $[sd]$ packet arrival rate to link ij is $r_{[sd]}(t) * V_{ij[sd]}(t)$, since all packets belonging to a virtual circuit are transmitted through the same link.

Let the packet service requirement be exponentially distributed with mean $1/\mu > 0$ and the service rate at link ij be $C_{ij} > 0$. Then the mean packet service time at link ij is $1/\mu_{ij} = 1/(\mu * C_{ij})$. If the network is also controlled by link-by-link error and window flow control, then we can derive the equivalent mean packet service time at link ij [137]. Packets are serviced according to first-come-first-served or processor sharing scheduling. Katevenis [247] and Morgan [333] preallocate buffer space to each virtual circuit in every node and multiplex packets from different (thousands) virtual circuits using round-robin scheduling. So, for the $[sd]$ packet process, we model each link ij either as an $M/M/1$ (Figure 5.7) or as a Processor Sharing queue (Figure 5.8), with packet arrival rate $r_{[sd]}(t) * V_{ij[sd]}(t)$ and mean service time $1/\mu_{ij}(t)$. Note, that for the Processor Sharing discipline, the packet service requirement may be generally distributed and packets from different classes of virtual circuits may have different mean service requirements.

Let $N_{ij[sd]}(t) \geq 0$ be the expected number of $[sd]$ packets at link ij at time t and $\mathbf{N}_{ij}(t) = [...N_{ij[sd]}(t)...]^T$ be the vector of the expected number of packets on link ij for all source-destination processes. Let $\rho_{ij[sd]}(\mathbf{N}_{ij}(t))$ be the probability that there is an $[sd]$ packet at link ij (either in queue or in transmission) at time t (call this probability: "instantaneous utilization for link ij for the $[sd]$ traffic"), such that the $[sd]$ packet departure rate from link ij at time t is $\mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t))$.

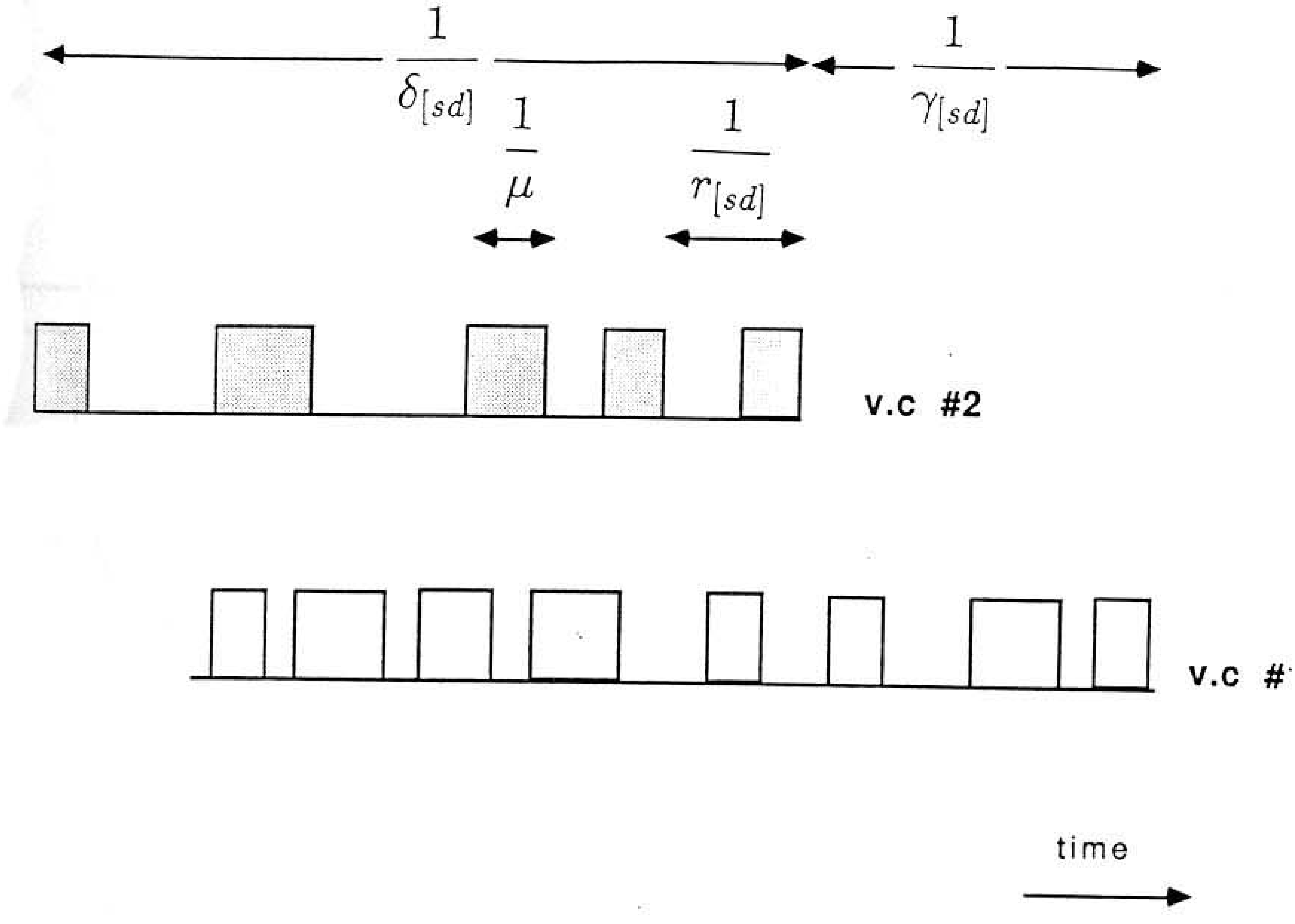

Figure 5.6: Two virtual circuits and their packets.

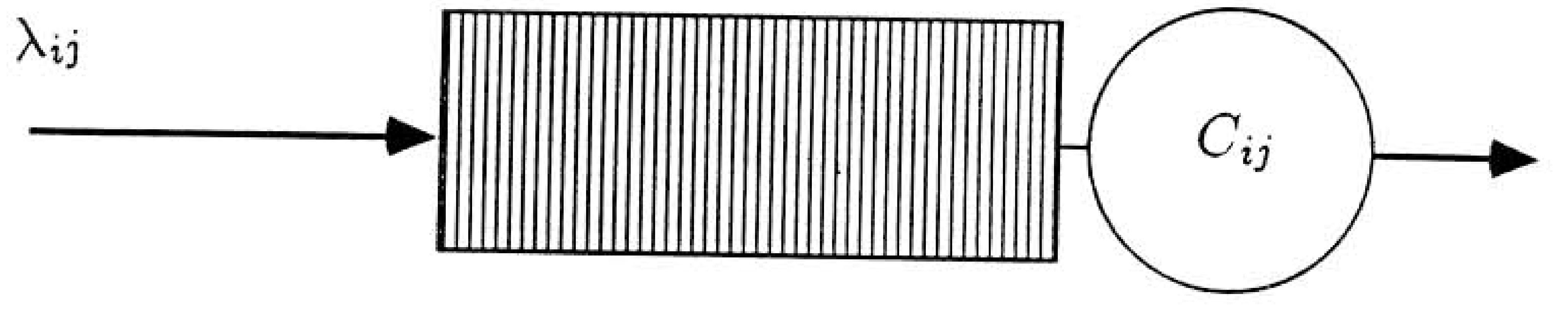

Figure 5.7: $M/M/1$ model for packet process.

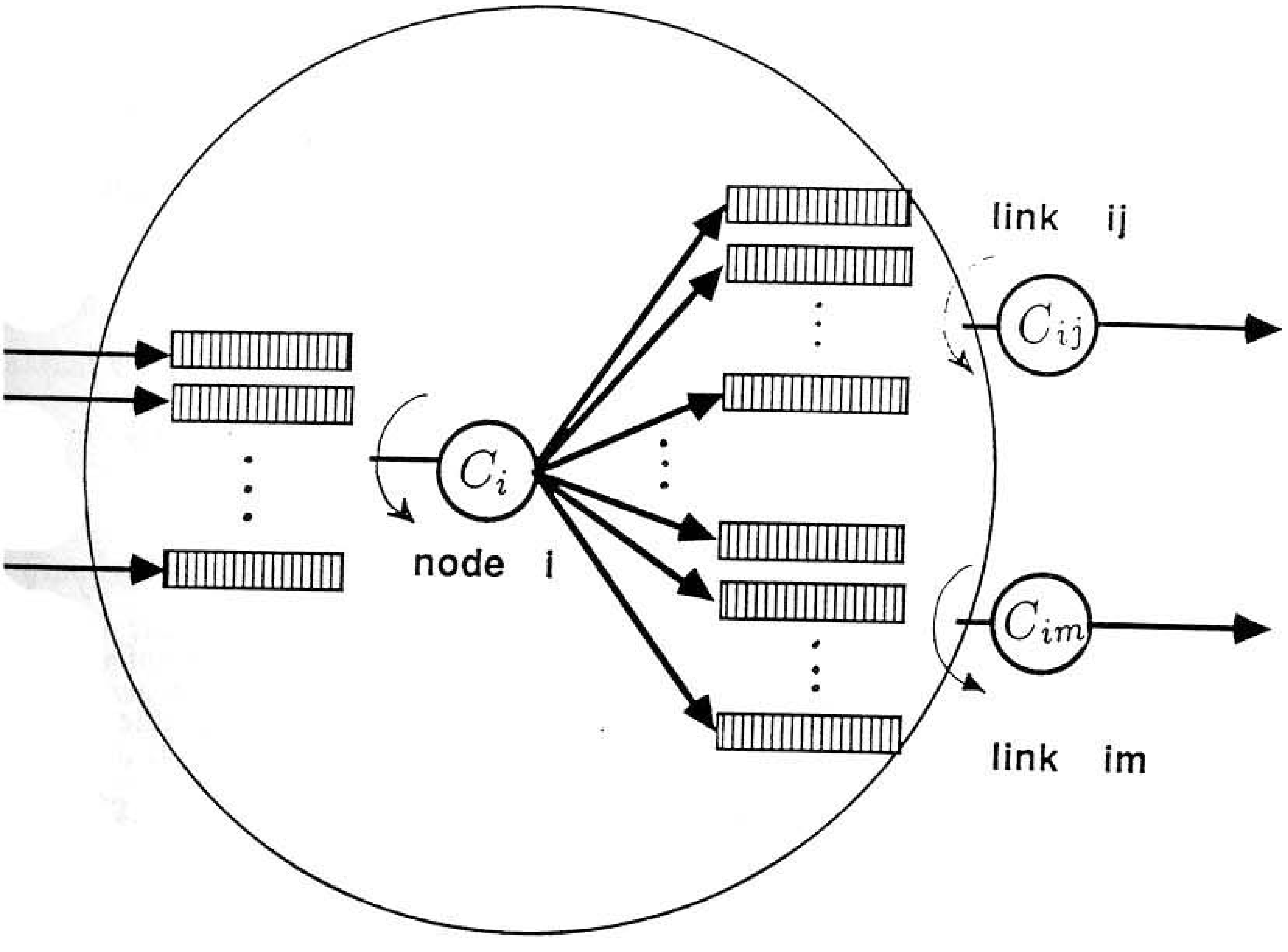

Figure 5.8: Processor Sharing model for packet process.

Then the expected number of $[sd]$ packets at link ij at time t, $N_{ij[sd]}(t)$, increases during Δt by the expected number of $[sd]$ packets that arrive during this period, $r_{[sd]}(t) * V_{ij[sd]}(t) * \Delta t$, minus the expected number of $[sd]$ packets that depart during this period, $\mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t))$. Since, the link utilization $\rho_{ij[sd]}(\mathbf{N}_{ij}(t))$, is a nonlinear function of the number of packets at link ij, $\mathbf{N}_{ij}(t)$, the $[sd]$ packet process at link ij is described by a nonlinear dynamic model

$$N_{ij[sd]}(t+\Delta t) = N_{ij[sd]}(t) + r_{[sd](t)} * V_{ij[sd]}(t) * \Delta t - \mu_{ij}(t) * \rho_{ij[sd]}(t)(\mathbf{N}_{ij}(t)) * \Delta t \quad \forall\, ij,\ [sd]$$

The expected number of $[sd]$ packets at link ij at time t, $N_{ij[sd]}(t)$, is a continuous function of time. So, let us define

$$\dot{N}_{ij[sd]}(t) = \lim_{\Delta t \to 0} \frac{N_{ij[sd]}(t+\Delta t) - N_{ij[sd]}(t)}{\Delta t} \qquad \forall\, ij \quad \forall\, [sd]$$

then the $[sd]$ packet process at link ij at time t is described by

$$\dot{N}_{ij[sd]}(t) = r_{[sd]}(t) * V_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t)) \qquad \forall\, ij \quad \forall\, [sd]$$

The state of the network is described by the expected number of virtual circuits $V_{ij[sd]}(t)$ and of packets $N_{ij[sd]}(t)$ for each link ij for each $[sd]$ traffic. So, we define the network state as

$$\mathbf{X}(t) = \begin{bmatrix} \cdots \\ V_{ij[sd]}(t) \\ N_{ij[sd]}(t) \\ \cdots \end{bmatrix}$$

The control variables are the congestion control parameters $\phi_{o[sd]}(t)$ and the routing fractions $\phi_{\pi[sd]}(t)$ for each path $\pi[sd]$, for each $[sd]$ traffic. So, let define the control vector for the whole network as

$$\mathbf{U}(t) = \begin{bmatrix} \cdots \\ \phi_{o[sd]}(t) \\ \cdots \\ \phi_{\pi[sd]}(t) \\ \cdots \end{bmatrix}$$

In order to write the dynamic evolution of the network state in vector form, we define the following auxiliary functions

$$f_{V,ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) - \delta_{[sd]}(t) * V_{ij[sd]}(t) \quad \forall\ ij,\ [sd]$$

$$f_{N,ij[sd]}(t) = r_{[sd]}(t) * V_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t)) \quad \forall\ ij,\ [sd]$$

$$\mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t)) = \begin{bmatrix} \cdots \\ f_{V,ij[sd]}(t) \\ f_{N,ij[sd]}(t) \\ \cdots \end{bmatrix}$$

Then the network dynamics are described by the following nonlinear differential equation

$$\dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t))$$

Finally, note that the $\phi_{o[sd]}$ and $\phi_{\pi[sd]}$'s represent the *fraction* of incoming flow to node s that is rejected or routed through path $\pi[sd]$. These fractions may be realized either with a *probabilistic* implementation or with a *deterministic* implementation, for example round-robin or thresholding. We discuss this further in section 1.2.

In this section, we have introduced a dynamic nonlinear queueing model for multi-class multi-destination virtual circuit networks. In the next section, we will use this nonlinear dynamic model to formulate and solve the combined routing and congestion control problem for dynamic virtual circuit networks as an optimal control problem.

5.6.3 Optimal Control Formulation

In this section, we formulate the joint routing and congestion control problem for multi-destination multi-class dynamic virtual circuit networks as an optimal control problem.

First, we define a multi-objective function $f(t, \mathbf{X}(t), \mathbf{U}(t))$ for the integrated problem. We would like to minimize the cost of rejecting virtual circuits from the network, of setting up and maintaining the virtual circuits inside the network, as well as of packet delay, while maximize the profit from servicing packets during a time interval $[t_0, t_f]$. To accomplish this, we define the following nonnegative costs and profits:

$C_{so[sd]}(t)$	: cost of not admitting a new $[sd]$ virtual circuit into the network at time t.
$C_{V,ij[sd]}(t)$	: cost per $[sd]$ virtual circuit for link ij at time t, for example the cost of setting up and maintaining the virtual circuit path through link ij.
$C_{N,ij[sd]}(t)$	: cost per $[sd]$ packet at link ij at time t.
$C_{\mu,ij[sd]}(t)$	: profit from servicing an $[sd]$ packet at link ij at time t.

So, given an initial time t_0 and a final time t_f, we define as our multi-objective function the following time-dependent function of the state $\mathbf{X}(t)$ and the controls $\mathbf{U}(t)$:

$$\begin{aligned} g(t, \mathbf{X}(t), \mathbf{U}(t)) = & \sum_{[sd]} C_{o[sd]}(t) * \gamma_{[sd]}(t) * \phi_{o[sd]}(t) + \\ & + \sum_{[sd]} \sum_{ij} C_{V,ij[sd]}(t) * V_{ij[sd]}(t) + \\ & + \sum_{[sd]} \sum_{ij} C_{N,ij[sd]}(t) * N_{ij[sd]}(t) - \\ & - \sum_{[sd]} \sum_{ij} C_{\mu,ij[sd]}(t) * \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t)) \end{aligned}$$

The first term of the objective function is the average loss of not admitting new virtual circuits into the network at every source node s for every $[sd]$ traffic. The second term is the average cost of setting up and maintaining $V_{ij[sd]}(t)$ virtual circuits on every link ij for every $[sd]$ traffic. The third term is the average cost of packet delay at every link ij for every $[sd]$ traffic. Finally, the last term is the profit from servicing an [sd] packet on every link ij.

Next, we define the set for the controls as

$$\mathbf{V} = \{\phi_{o[sd]}(t),\ \phi_{\pi[sd]}(t)\ \ \forall\ \pi[sd]\ \ \forall\ [sd], \quad such\ that\ \ forall\ [sd]$$

$$\phi_{o[sd]}(t) \geq 0,\ \ \phi_{\pi[sd]}(t) \geq 0\ \ \forall\ \pi[sd],\ \ \phi_{o[sd]}(t) + \sum_{\pi[sd]} \phi_{\pi[sd]}(t) = 1\}$$

Nonnegative constraints on the network state $V_{ij[sd]}(t) \geq 0$ and $N_{ij[sd]}(t) \geq 0$ are always satisfied due to the structure of $\mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t))$.

Define also $P_{V,ij[sd]}(t)$ to be the costate variable for $V_{ij[sd]}(t)$, the expected number of $[sd]$ virtual circuits on link ij, and $P_{N,ij[sd]}(t)$ to be the costate variable for $N_{ij[sd]}(t)$, the expected number of $[sd]$ packets on link ij. Then the costate variable vector for all links ij for all $[sd]$ processes is $\mathbf{P}(t) = [\ldots P_{V,ij[sd]}(t)\ \ P_{N,ij[sd]}(t) \ldots]^T$.

Then our Dynamic Virtual Circuit Routing and Congestion Control problem *(DVCRCC)* is:

Problem DVCRCC:

minimize	$\int_{t_0}^{t_f} g(t, \mathbf{X}(t), \mathbf{U}(t))dt$
with respect to	$\mathbf{U}(t)$
such that	$\dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t))$
	$\mathbf{X}(t_0) = \mathbf{X}_0$
	$\mathbf{X}(t_f)\ free$
	$\mathbf{U}(t) \in \mathbf{V}$

where

t_0	fixed initial time,
t_f	fixed final time,
$\mathbf{X}(t)$	network state,
$\mathbf{U}(t)$	controls,
$g(t, \mathbf{X}(t), \mathbf{U}(t))$	objective function,
$\mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t))$	state dynamics,
$\mathbf{V}$	control set,
$\mathbf{X}(t_0) = \mathbf{X}_0$	initial network state,
$\mathbf{X}(t_f)$	final network state,

The Hamiltonian function of the state $\mathbf{X}(t)$, the controls $\mathbf{U}(t)$ and the costate variables $\mathbf{P}(t)$ at time t is

$$H(t, \mathbf{X}(t), \mathbf{U}(t), \mathbf{P}(t)) \;=\; g(t, \mathbf{X}(t), \mathbf{U}(t)) + \mathbf{P}(t) * \mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t), \mathbf{P}(t))$$

Note that the objective function g in the Hamiltonian has a multiplier equal to 1, since we have free final state conditions.

Necessary conditions for optimality are provided by Pontryagin's maximum principle [79, 33, 58, 162, 502].

Theorem 1. *Necessary conditions*

Let $\mathbf{U}^*(t)$ be a piecewise continuous control defined on $[t_0, t_f]$ which solves Problem DVCRCC and let $\mathbf{X}^*(t)$ be the associated optimal path. Then there exists a continuous and piecewise continuously differentiable vector function $\mathbf{P}(t) = [...P_{V,ij[sd]}(t)\ P_{N,ij[sd]}(t)...]^T$ such that the following conditions are satisfied for all $t \in [t_0, t_f]$,

$$\phi^*_{o[sd]}(t) \begin{cases} > 0 & only\ if\ C_{o[sd]}(t) = \min\{C_{o[sd]}, \min\limits_{p[sd]}\{\sum\limits_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & o.w.\ \ \forall\ [sd] \end{cases}$$

$$\phi^*_{\pi[sd]}(t) \begin{cases} > 0 & only\ if \sum\limits_{ij} P_{V,ij[sd]}(t) * 1_{ij \in \pi[sd]}(t) = \\ & = \min\{C_{o[sd]}(t), \min\limits_{p[sd]}\{\sum\limits_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & o.w.\ \ \forall\ \pi[sd]\ \ \forall\ [sd] \end{cases}$$

$$\dot{V}^*_{ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi^*_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) - \delta_{[sd]}(t) * V^*_{ij[sd]}(t) \quad \forall\ ij,\ [sd]$$

$$\dot{N}^*_{ij[sd]}(t) = r_{[sd]}(t) * V^*_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}^*_{ij}(t)) \quad \forall\ ij,\ [sd]$$

$$V^*_{ij[sd]}(t_0) = V_{ij[sd],0} \quad \forall\ ij,\ [sd]$$
$$N^*_{ij[sd]}(t_0) = N_{ij[sd],0} \quad \forall\ ij,\ [sd]$$

$$\dot{P}_{V,ij[sd]}(t) = -\{\ C_{V,ij[sd]}(t) - P_{V,ij[sd]}(t) * \delta_{[sd]}(t) + P_{N,ij[sd]}(t) * r_{[sd]}(t)\ \} \quad \forall\ ij \quad \forall\ [sd]$$

$$\dot{P}_{N,ij[sd]}(t) = -\{\ C_{N,ij[sd]}(t) - \sum_{[s_1d_1]} C_{\mu,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)} - - \sum_{[s_1d_1]} P_{N,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)}\ \} \quad \forall\ ij \quad \forall\ [sd]$$

$$P_{V,ij[sd]}(t_f) = 0 \quad \forall\ ij \quad \forall\ [sd]$$
$$P_{N,ij[sd]}(t_f) = 0 \quad \forall\ ij \quad \forall\ [sd]$$

Proof: The Hamiltonian must satisfy the following condition

$H(t,\mathbf{X}^*(t),\mathbf{U}^*(t),\mathbf{P}(t)) \leq H(t,\mathbf{X}^*(t),\mathbf{U},\mathbf{P}(t)) \quad \forall\ \mathbf{U} \in \mathbf{V}$

which is equivalent to the following condition

$$\sum_{[sd]} \left\{ \gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi^*_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi^*_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t)] \right\} \leq$$
$$\leq \sum_{[sd]} \left\{ \gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t)] \right\}$$
$\forall\ \phi_{o[sd]},\ \phi_{\pi[sd]} \in \mathbf{V} \quad \forall\ \pi[sd]\ \forall\ [sd]$

Since, there is no dependency among the controls for different source-destination pairs $[sd]$, we can decomposed the above conditions $\forall\ [sd]$ to

$$\gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi^*_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi^*_{ij[sd]}(t) * 1_{ij\in\pi[sd]}(t)] \leq$$

$$\leq \gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi_{ij[sd]}(t) * 1_{ij \in \pi[sd]}(t)]$$

$\forall\ \phi_{o[sd]},\ \phi_{\pi[sd]} \in \mathbf{V} \quad \forall\ \pi[sd]$

Then the optimal controls satisfy the following conditions

$$\phi^*_{o[sd]}(t) \begin{cases} > 0 & \text{only if } C_{o[sd]}(t) = \min\{C_{o[sd]}(t), \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & \text{o.w. } \forall\ [sd] \end{cases}$$

$$\phi^*_{\pi[sd]}(t) \begin{cases} > 0 & \text{only if } \sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in \pi[sd]}(t) = \\ & = \min\{C_{o[sd]}(t), \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & \text{o.w. } \forall\ \pi[sd]\ \ \forall\ [sd] \end{cases}$$

The optimal state and control pair $(\mathbf{X}^*(t), \mathbf{U}^*(t))$ must also satisfy the state dynamics

$\dot{\mathbf{X}}^*(t) = \mathbf{f}(t, \mathbf{X}^*(t), \mathbf{U}^*(t))$

which can be rewritten as

$$\dot{V}^*_{ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi^*_{\pi[sd]}(t) * 1_{ij \in \pi[sd]}(t) - \delta_{[sd]}(t) * V^*_{ij[sd]}(t) \quad \forall\ ij,\ [sd]$$

$$\dot{N}^*_{ij[sd]}(t) = r_{[sd]}(t) * V^*_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}^*_{ij}(t)) \quad \forall\ ij,\ [sd]$$

The optimal state must also satisfy the initial state $\mathbf{X}^*(t_0) = \mathbf{X}_0$, therefore

$$\begin{aligned} V^*_{ij[sd]}(t_0) &= V_{ij[sd],0} \quad \forall\ ij,\ [sd] \\ N^*_{ij[sd]}(t_0) &= N_{ij[sd],0} \quad \forall\ ij,\ [sd] \end{aligned}$$

The costate variables must satisfy the following conditions

$\dot{\mathbf{P}}(t) = -\nabla_{\mathbf{X}} H(t, \mathbf{X}^*(t), \mathbf{U}^*(t), \mathbf{P}(t))$

which can be rewritten as

$$\dot{P}_{V,ij[sd]}(t) = -\frac{\partial H(t,\mathbf{X}^*(t),\mathbf{U}^*(t),\mathbf{P}(t))}{\partial V_{ij[sd]}(t)} =$$

$$= -\{\ C_{V,ij[sd]}(t) - P_{V,ij[sd]}(t) * \delta_{[sd]}(t) + P_{N,ij[sd]}(t) * r_{[sd]}(t)\ \}$$

$$\dot{P}_{N,ij[sd]}(t) = -\frac{\partial H(t,\mathbf{X}^*(t),\mathbf{U}^*(t),\mathbf{P}(t))}{\partial N_{ij[sd]}(t)} =$$

$$= -\{\ C_{N,ij[sd]}(t) - \sum_{[s_1d_1]} C_{\mu,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)} -$$

$$-\sum_{[s_1d_1]} P_{N,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)}\ \} \quad \forall\ ij,\ [sd]$$

Since we have no conditions on the final state $\mathbf{X}(t_f)$, the costate variables at the final time must be zero, $\mathbf{P}(t_f) = 0$. Therefore

$P_{V,ij[sd]}(t_f) = 0 \quad \forall\ ij\ \ \forall\ [sd]$

$P_{N,ij[sd]}(t_f) = 0 \quad \forall\ ij\ \ \forall\ [sd]$ □

Sufficient conditions for optimality are provided by the convexity of the Hamiltonian with respect to the state and the controls [320, 237, 439, 379].

Theorem 2. *Sufficient conditions*

Let $(\bar{\mathbf{X}}(t), \bar{\mathbf{U}}(t))$ be an admissible pair in Problem DVCRCC. Assume that $\rho_{ij[sd]}(\mathbf{N}_{ij}(t))$ is defined for $\mathbf{N}_{ij}(t) \geq 0$, is concave monotonically increasing and twice differentiable in $\mathbf{N}_{ij}(t)$. If there exists a continuous and piecewise continuously differentiable vector function $\mathbf{P}(t) = [\ldots P_{V,ij[sd]}(t)\ P_{N,ij[sd]}(t)\ldots]^T$ such that the following conditions are satisfied for all $t \in [t_0, t_f]$

$$\bar{\phi}_{o[sd]}(t)\begin{cases} > 0 & \textit{only if } C_{o[sd]}(t) = \min\{C_{o[sd]}(t), \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij\in p[sd]}(t)\}\} \\ = 0 & o.w.\ \ \forall\ [sd] \end{cases}$$

$$\bar{\phi}_{\pi[sd]}(t) \begin{cases} > 0 & only\ if \sum_{ij} P_{V,ij[sd]}(t) * 1_{ij\in\pi[sd]}(t) = \\ & = \min\{C_{o[sd]}(t), \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij\in p[sd]}(t)\}\} \\ = 0 & o.w.\ \ \forall\ \pi[sd]\ \ \forall\ [sd] \end{cases}$$

$$\dot{\bar{V}}_{ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \bar{\phi}_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) - \delta_{[sd]}(t) * \bar{V}_{ij[sd]}(t) \quad \forall\ ij,\ [sd]$$

$$\dot{\bar{N}}_{ij[sd]}(t) = r_{[sd]}(t) * \bar{V}_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\bar{\mathbf{N}}_{ij}(t)) \quad \forall\ ij,\ [sd]$$

$$\bar{V}_{ij[sd]}(t_0) = V_{ij[sd],0} \quad \forall\ ij\ \ \forall\ [sd]$$
$$\bar{N}_{ij[sd]}(t_0) = N_{ij[sd],0} \quad \forall\ ij\ \ \forall\ [sd]$$

$$\dot{P}_{V,ij[sd]}(t) = -\{\ C_{V,ij[sd]}(t) - P_{V,ij[sd]}(t) * \delta_{[sd]}(t) + P_{N,ij[sd]}(t) * r_{[sd]}(t)\ \}$$

$$\dot{P}_{N,ij[sd]}(t) = -\{\ C_{N,ij[sd]}(t) - \sum_{[s_1d_1]} C_{\mu,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\bar{\mathbf{N}}_{ij}(t))}{dN_{ij[sd]}(t)} -$$
$$- \sum_{[s_1d_1]} P_{N,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\bar{\mathbf{N}}_{ij}(t))}{dN_{ij[sd]}(t)}\ \} \quad \forall\ ij,\ [sd]$$

$$P_{N,ij[sd]}(t) \geq 0 \quad \forall\ ij\ \ \forall\ [sd]$$

$$P_{V,ij[sd]}(t_f) = 0 \quad \forall\ ij\ \ \forall\ [sd]$$
$$P_{N,ij[sd]}(t_f) = 0 \quad \forall\ ij\ \ \forall\ [sd]$$

then $(\bar{\mathbf{X}}(t), \bar{\mathbf{U}}(t))$ *is optimal.*

Proof: The first part of the proof is similar to that of Theorem 1.

In addition, the control set $\mathbf{V}$ is a convex set and since $-\rho_{ij[sd]}(\mathbf{N}_{ij}(t))$ is a convex (i.e. $\rho_{ij[sd]}(\mathbf{N}_{ij}(t))$ is concave) and differentiable function in $\mathbf{N}_{ij}(t)$, our objective function $g(t, \mathbf{X}(t), \mathbf{U}(t))$, as well as each component of $\mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t))$

are differentiable and convex functions in the variables $(\mathbf{X}(t), \mathbf{U}(t))$ for $t \in [t_0, t_f]$. Furthermore, if $P_{N,ij[sd]}(t) \geq 0 \quad \forall\ ij \quad \forall\ [sd]$, then the Hamiltonian function $H(t, \mathbf{X}(t), \mathbf{U}(t), \mathbf{P}(t))$ is a convex function in $(\mathbf{X}(t), \mathbf{U}(t))$ for $t \in [t_0, t_f]$ (we need nonnegativity of the costate variables only for those components of $\mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t))$ that are nonlinear in $\mathbf{X}(t)$ [320, 237, 439, 379]).

If all the above conditions are satisfied, then $(\bar{\mathbf{X}}(t), \bar{\mathbf{U}}(t))$ is optimal. □

Note that for an $M/M/1$ or Processor Sharing queue at steady state, $\rho_{ij[sd]}(\mathbf{N}_{ij}) = \dfrac{N_{ij[sd]}}{1 + \sum_{[s_1 d_1]} \rho_{ij[s_1 d_1]}}$ is defined for $\mathbf{N}_{ij} \geq 0$, is concave, monotonically increasing and twice differentiable in $\mathbf{N}_{ij}$ with $\lim_{\mathbf{N}_{ij} \to \infty} \rho_{ij}(\mathbf{N}_{ij}) = 1$.

So, after numerically solving a two-Point Boundary-Value Problem (TPBVP), we have the optimal congestion control and routing decisions. Numerical methods [14, 79, 203, 254, 292, 415] for the solution of such problems involve either flooding or iterative procedures. *Flooding* (or dynamic programming) procedures start from a point that satisfies one boundary condition and generates a trajectory. This is repeated many times until one of these trajectories satisfies the other condition or an interpolation of these trajectories can give an acceptable solution. *Iterative* procedures use successive linearization. A nominal solution is chosen such that to satisfy one or more of the following conditions: 1) state differential equations, 2) adjoint differential equations, 3) optimality conditions, 4) boundary conditions. Then this nominal solution is modified by successive linearization such that the remaining conditions are also satisfied. Three classes of iterative procedures may be used: i) *neighboring extremal*, ii) *gradient*, and iii) *quasi-linearization* procedures.

In this section, we are primarily interested in the optimal control formulation for the finite horizon problem and the long-run stationary equilibrium solution. So, we will not discuss further numerical techniques for the finite horizon optimal control problem. In this section, we formulated the combined routing and congestion control problem for multi-destination multi-class dynamic virtual circuit networks as an optimal control problem. Then for specific network configuration and traffic characteristics, we can find the optimum congestion control and routing decisions by solving a TPBVP. We can decompose the above problem to many smaller subproblems, one for every source-destination. However, numerical solution may require long computational times for on line implementation. Therefore, in the next section, we also derive state dependent routing and congestion controls for the long-run stationary equilibrium that can be used for on-line implementation.

5.6.4 State Dependent Routing & Congestion Control

In this section, we consider a network with constant arrival rates and mean durations of virtual circuits, as well as constant costs and profits (autonomous system), and we find optimal state dependent virtual circuit routing and congestion controls for the long-run stationary equilibrium. Our problem becomes

$$
\begin{aligned}
\textit{minimize} \qquad & \sum_{[sd]} C_{o[sd]} * \gamma_{[sd]} * \phi_{o[sd]} + \\
+ \quad & \sum_{[sd]} \sum_{ij} C_{V,ij[sd]} * V_{ij[sd]} + \\
+ \quad & \sum_{[sd]} \sum_{ij} C_{N,ij[sd]} * N_{ij[sd]} - \\
- \quad & \sum_{[sd]} \sum_{ij} C_{\mu,ij[sd]} * \mu_{ij} * \rho_{ij[sd]}(\mathbf{N}_{ij})
\end{aligned}
$$

$$\begin{array}{lll} \textit{with respect to} & \text{the congestion controls } \phi_{o[sd]} \geq 0 & \forall\ [sd] \\ & \text{the routing fractions } \phi_{\pi[sd]} \geq 0 & \forall\ \pi[sd]\ \ \forall\ [sd] \end{array}$$

$$\textit{such that} \qquad 0 = \sum_{\pi[sd]} \gamma_{[sd]} * \phi_{\pi[sd]} * 1_{ij \in \pi[sd]} - \delta_{[sd]} * V_{ij[sd]} \qquad \forall\ ij\ \ \forall\ [sd]$$

$$0 = r_{[sd]} * V_{ij[sd]} - \mu_{ij} * \rho_{ij[sd]}(\mathbf{N}_{ij}) \qquad \forall\ ij\ \ \forall\ [sd]$$

$$\phi_{o[sd]},\ \ \phi_{\pi[sd]} \geq 0 \qquad \forall\ \pi[sd]\ \ \forall\ [sd]$$

$$\phi_{o[sd]} + \sum_{\pi[sd]} \phi_{\pi[sd]} = 1 \qquad \forall\ [sd]$$

The minimization of the Hamiltonian with respect to the congestion control and routing fractions is equivalent to the following minimization problem

$$\textit{minimize} \sum_{[sd]} \left\{ \gamma_{[sd]} * \left[C_{o[sd]} * \phi_{o[sd]} + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]} * \phi_{\pi[sd]} * 1_{ij \in \pi[sd]}\right] \right\}$$

$$\textit{with respect to} \qquad \phi_{o[sd]}, \phi_{\pi[sd]}, \quad \forall\ \pi[sd],\ [sd]$$

$$\textit{such that} \qquad \phi_{o[sd]} + \sum_{\pi[sd]} \phi_{\pi[sd]} = 1 \quad \phi_{o[sd]},\ \phi_{\pi[sd]} \geq 0 \quad \forall\ \pi[sd] \quad \forall\ [sd]$$

where the costate variables $P_{V,ij[sd]}$ for the expected number of virtual circuits and the costate variables $P_{N,ij[sd]}$ for the expected number of packets for each link ij, for each $[sd]$ pair will be found later.

The above problem can be decomposed for each source-destination pair $[sd]$ to the following problem

$$\text{minimize} \qquad \gamma_{[sd]} * [C_{o[sd]} * \phi_{o[sd]} + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]} * \phi_{ij[sd]} * 1_{ij \in \pi[sd]}]$$

$$\text{with respect to} \qquad \phi_{o[sd]}, \phi_{\pi[sd]} \quad \forall \ \pi[sd]$$

$$\text{such that} \qquad \phi_{o[sd]} + \sum_{\pi[sd]} \phi_{\pi[sd]} = 1, \quad \phi_{o[sd]}, \ \phi_{\pi[sd]} \geq 0 \quad \forall \ \pi[sd]$$

Define the minimum cost at source node s for the $[sd]$ virtual circuit traffic to be $P^*_{V,s[sd]} = \min\{C_{o[sd]}, \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]} * 1_{ij \in p[sd]}\}\}$. Then the optimum congestion controls are:

$$\phi^*_{o[sd]} \begin{cases} > 0 & \text{only if } C_{o[sd]} = P^*_{V,s[sd]} \\ = 0 & \text{o.w.} \end{cases}$$

and the optimum routing fractions are:

$$\phi^*_{\pi[sd]} \begin{cases} > 0 & \text{only if } \sum_{ij} P_{V,ij[sd]} * 1_{ij \in \pi[sd]} = P^*_{V,s[sd]} \\ = 0 & \text{o.w.} \end{cases}$$

Therefore, an $[sd]$ virtual circuit is rejected at source node s only if the cost of rejecting it is equal to the minimum cost at node i, i.e. $C_{o[sd]} = P^*_{V,s[sd]}$. Also, path $\pi[sd]$ will be used for the $[sd]$ traffic only if its costate variable achieves the minimum cost, i.e. $\sum_{ij} P_{V,ij[sd]} * 1_{ij \in \pi[sd]} = P^*_{V,s[sd]}$.

When the congestion control and routing fractions achieve their optimum values, we have

$$C_{o[sd]} * \phi^*_{o[sd]} + \sum_{ij} \sum_{\pi[sd]} P_{V,ij[sd]} * \phi^*_{ij[sd]} * 1_{ij \in \pi[sd]} = P^*_{V,s[sd]}$$

The optimum congestion control and routing decisions depend on the values of the costate variables $P_{V,ij[sd]}$ $\forall \ ij$ $\forall \ [sd]$. So, we have to calculate the costate variables $P_{V,ij[sd]}$ for each link ij for each $[sd]$ traffic.

At steady state, the costate variables must satisfy $\dot{P}_{V,ij[sd]} = 0 \quad \forall\ ij,\ [sd]$.

$\dot{P}_{V,ij[sd]} = 0 \Rightarrow \quad -\{\ C_{V,ij[sd]} - P_{V,ij[sd]} * \delta_{[sd]} + P_{N,ij[sd]} * r_{[sd]}\ \} = 0 \quad \forall\ ij,\ [sd]$

Then

$$P_{V,ij[sd]} \ = \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * P_{N,ij[sd]} \quad \forall\ ij \quad \forall\ [sd]$$

Next, in order to calculate the costate variables $P_{V,ij[sd]}$ for the expected number of $[sd]$ virtual circuits, we must first calculate the costate variables $P_{N,ij[sd]}$ for the expected number of $[sd]$ packets. At steady state, the costate variables must satisfy $\dot{P}_{N,ij[sd]} = 0 \quad \forall\ ij \quad \forall\ [sd]$.

$$\dot{P}_{N,ij[sd]} = 0 \Rightarrow \quad -\{\ C_{N,ij[sd]} - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}} -$$

$$- \sum_{[s_1 d_1]} P_{N,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}}\ \} = 0 \quad \forall\ ij \quad \forall\ [sd]$$

In order to find the costate variables $P_{N,ij[sd]}$ for the expected number of $[sd]$ packets on every link ij, we must solve a system of equations for all source-destination processes that use this link:

$$C_{N,ij[sd]} - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}} -$$

$$- \sum_{[s_1 d_1]} P_{N,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}} = 0 \quad \forall\ [sd]$$

Note that for an $M/M/1$ or Processor Sharing queueing model the expected number of $[sd]$ packets on link ij at steady state is

$$N_{ij[sd]} = \frac{\rho_{ij[sd]}}{1 - \sum_{[s_1 d_1]} \rho_{ij[s_1 d_1]}} \quad \forall\ [sd]$$

Solving the above system of equations (for all $[sd]$ traffic that use link ij), we have the utilization of link ij for each $[sd]$ process at steady state

$$\rho_{ij[sd]} = \frac{N_{ij[sd]}}{1 + \sum_{[s_1 d_1]} N_{ij[s_1 d_1]}}$$

Therefore we can rewrite the $P_{N,ij[sd]}$ costate variable system of equations for each link ij, as

$$C_{N,ij[sd]} - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]} * \mu_{ij}$$

$$* \left\{ \frac{1 + \sum_{[s_2 d_2] \neq [sd]} N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} \sum_{[s_2 d_2] \neq [sd]} \frac{N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} \right\} -$$

$$- \sum_{[s_1 d_1]} P_{N,ij[s_1 d_1]} * \mu_{ij} * \left\{ \frac{1 + \sum_{[s_2 d_2] \neq [sd]} N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} - \sum_{[s_2 d_2] \neq [sd]} \frac{N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} \right\} = 0$$

$\forall\ [sd]$

The solution to the above system is

$$P_{N,ij[sd]} = \frac{1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]}}{\mu_{ij}} * \Bigg[C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) +$$

$$+ \sum_{[s_2 d_2] \neq [sd]} C_{N,ij[s_2 d_2]} * N^*_{ij[s_2 d_2]} \Bigg] - C_{\mu,ij[sd]} \quad \forall\ [sd]$$

In the previous section, we stated that it must hold $P_{N,ij[sd]} \geq 0$. So, we must have

$$\frac{C_{N,ij[sd]}}{\mu_{ij}} - C_{\mu,ij[sd]} \geq 0 \quad \forall\ [sd]$$

i.e. the mean packet delay cost should be greater or equal to the profit from servicing this packet.

Note that for the special case of equal packet cost for the different $[sd]$ processes that use link ij, $C_{N,ij[s_2d_2]} = C_{N,ij} \ \forall \ [s_2d_2]$, the above solution simplifies to

$$P_{N,ij[sd]} = \frac{C_{N,ij} * (1 + \sum_{[s_1d_1]} N^*_{ij[s_1d_1]})^2}{\mu_{ij}} - C_{\mu,ij[sd]} \quad \forall \ [sd]$$

Substituting the $P_{N,ij[sd]}$ into $P_{V,ij[sd]}$, we have the cost to go from node s to destination d through path $\pi[sd]$:

$$\mathrm{P}_{V,\pi[sd]} = \sum_{ij\in\pi[sd]} \left\{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \left[\frac{1 + \sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} * \left[C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) + \right.\right.\right.$$

$$\left.\left.\left. + \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \right] - C_{\mu,ij[sd]} \right] \right\} \quad \forall \ \pi[sd] \quad \forall \ [sd]$$

The following Theorems follow immediately:

Theorem 3. <u>*Congestion Control*</u>

For the long-run stationary equilibrium of the virtual circuit routing and congestion control problem, at every source node s, for every destination node d, $[sd]$ virtual circuits are rejected at a node s only if the cost of rejecting them is less than the minimum cost to go from node s to the destination d through any of the paths $\pi[sd]$

$\phi^*_{o[sd]} > 0$ *only if*

$$C_{o[sd]} < \min_{p[sd]} \left\{ \sum_{ij\in p[sd]} \left\{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \left[\frac{1 + \sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} * \right] \right.\right.$$

$$\left.\left. * \left[C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) + \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \right] - C_{\mu,ij[sd]} \right] \right\} \right\}$$

Theorem 4. *Routing Rule*

For the long-run stationary equilibrium of the virtual circuit routing and congestion control problem, [sd] virtual circuits are routed through path $\pi[sd]$ only if the minimum cost to reach the destination d through path $\pi[sd]$ is the minimum

$\phi^*_{\pi[sd]} > 0$ *only if*

$$\sum_{ij\in\pi[sd]} \left\{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \left[\frac{1+\sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} \left[C_{N,ij[sd]} * (1+N^*_{ij[sd]}) + \right.\right.\right.$$

$$\left.\left.\left. + \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \right] - C_{\mu,ij[sd]} \right] \right\} =$$

$$= \min_{p[sd]} \left\{ C_{o[sd]}, \sum_{ij\in p[sd]} \left\{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \left[\frac{1+\sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} * \left[C_{N,ij[sd]} * (1+N^*_{ij[sd]}) + \right.\right.\right.\right.$$

$$\left.\left.\left.\left. + \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \right] - C_{\mu,ij[sd]} \right] \right\} \right\}$$

From the above analysis, we have derived that the length of each link ij for an $[sd]$ virtual circuit is

$$l_{ij[sd]} = \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \left[\frac{1+\sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} \left[C_{N,ij[sd]} * (1+N^*_{ij[sd]}) + \right.\right.$$

$$\left.\left. + \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \right] - C_{\mu,ij[sd]} \right]$$

The first term of the above length represents the cost $C_{V,ij[sd]}$ for setting up and maintaining an $[sd]$ virtual circuit passing through link ij, times the average virtual circuit duration $1/\delta_{[sd]}$. The second term of the above length represents the average number of packets $r_{[sd]}/\delta_{[sd]}$ in this $[sd]$ virtual circuit times the cost $C_{N,ij[sd]}$ per packet. Finally, the last term of the above link length represents the profit $C_{\mu,ij[sd]}$ from servicing an $[sd]$ packet on link ij

Let us consider the special case where we have zero $[sd]$ virtual circuit set up and maintenance cost $C_{V,ij[sd]} = 0$, zero profit $C_{\mu,ij[sd]} = 0$ for servicing $[sd]$ packets, and unit delay costs $C_{N,ij[s_2d_2]} = 1$, $\forall$ $[s_2d_2]$ on link ij. Then the length of link ij for an $[sd]$ virtual circuit is

$$l_{ij[sd]} = \frac{r_{[sd]}}{\delta_{[sd]}} * \frac{(1 + \sum_{[s_1d_1]} N^*_{ij[s_1d_1]})^2}{\mu_{ij}}$$

That means, that when our only objective is to minimize the average packet delay, then the link length is given by a quadratic function of the average number of packets on this link.

In this section, we have derived state dependent routing and congestion controls for multi-class multi- destination virtual circuit networks. In the next section, we investigate a simple case of this state dependent routing algorithm via simulation.

5.6.5 Simulation

In this section, we investigate a simple case of the derived state dependent virtual circuit routing algorithm via simulation. Simulation is a very effective model for a detailed investigation of the routing algorithm. While the analytical models provide a rigorous mathematical analysis of the system, they cannot afford too much complexity. If we try to include all the parameters that affect the system, then the analytical model become intractable. On the other hand, simulation models are computer programs [289, 288, 149, 428, 316, 351, 408, 314] and therefore we can program as much detail as we like. Their drawback is that they are time consuming. We have to spend a lot of time for writing the code as well as for running them. However, simulation is the only way (besides real implementation) to measure the performance of dynamic routing algorithms.

We have implemented three deterministic source routing algorithms along the minimum length path for single class virtual circuit networks (ties are broken arbitrarily - though we seldom have ties). The first algorithm uses as link length a special case of that proposed in the previous section. The second algorithm uses as link length the expected packet delay on this link. Finally, the third algorithm is the optimal quasi-static routing algorithm.

1) *Quadratic routing*:

send a new virtual circuit along path π,

$$\text{if} \sum_{ij} \frac{(1+N_{ij})^2}{\mu_{ij}} * 1_{ij\in\pi} = \min_p \left\{ \sum_{ij} \frac{(1+N_{ij})^2}{\mu_{ij}} * 1_{ij\in p} \right\}$$

2) *Shortest queue routing* :

send a new virtual circuit along path π,

$$\text{if} \sum_{ij} \frac{1+N_{ij}}{\mu_{ij}} * 1_{ij\in\pi} = \min_p \left\{ \sum_{ij} \frac{1+N_{ij}}{\mu_{ij}} * 1_{ij\in p} \right\}$$

3) *Optimal quasi-static routing*

For updating the information at the source node about the link lengths in the network, we considered three factors:

1) *what estimate* of the number of packets N_{ij} at each link ij is sent to the source node from each node i.

2) *how often* this estimate is sent to the source node by each node i. It is well known that the updating period should be smaller than the average virtual circuit duration [177, 518].

3) *after how much delay* this information arrives back to the source node. We assume that no extra traffic is created from each node to the source node, but that this information is either piggybacked on other packets or it is transferred through a different channel.

First, we consider a single source-destination network with 2 paths from source to destination (Figure 5.9) that have the same capacity but the order of their links is different.

Path # has 7 links with transmission rates 5, 4, 3, 3, 2, 1 and 1. Path #2 has 7 links with transmission rates 1, 1, 2, 3, 3, 4 and 5.

The mean packet service time is $\frac{1}{\mu} = 1$ and therefore $\mu_{ij} = \mu * C_{ij} = C_{ij}$. The mean virtual circuit duration is $\frac{1}{\delta} = 1000$. The total packet arrival rate is

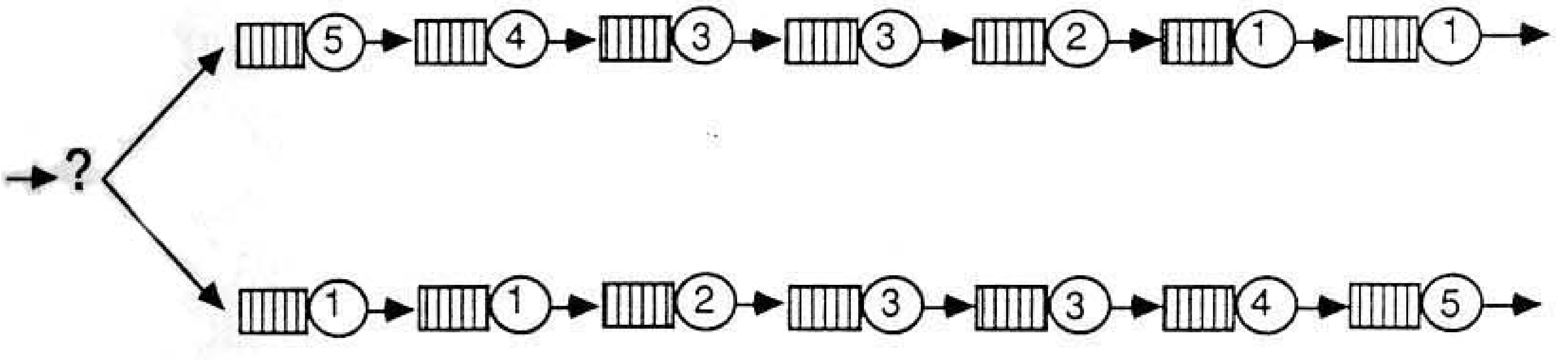

Figure 5.9: Simulated network.

$r * \frac{\gamma}{\delta} = \frac{1000}{700}$, however we considered 5 cases that achieve this total packet arrival rate:

γ	r	$\frac{\gamma}{\delta}$	$\frac{r}{\delta}$
$\frac{1}{7}$	$\frac{1}{100}$	$\frac{1000}{7}$	10
$\frac{1}{14}$	$\frac{1}{50}$	$\frac{1000}{14}$	20
$\frac{1}{26}$	$\frac{1}{27}$	$\frac{1000}{26}$	$\frac{1000}{27}$
$\frac{1}{50}$	$\frac{1}{14}$	20	$\frac{1000}{14}$
$\frac{1}{100}$	$\frac{1}{7}$	10	$\frac{1000}{7}$

where γ is the arrival rate of virtual circuits, r is the packet arrival rate per virtual circuit, $\frac{\gamma}{\delta}$ is the average number of virtual circuits into the network and $\frac{r}{\delta}$ is the average number of packets per virtual circuit.

The information at the source node about the link lengths in the network is updated according to two schemes:

a) *instantaneous* information, when at every instant, the source node knows and uses the current number of packets at every link and

b) *obsolete* information, when the information about the average number of packets at every link during a time interval of 100 time units is sent to the source node at the end of this time interval and it is used by the source node after 50 time units delay.

Figures 5.10, 5.11, 5.12, 5.13, 5.14 and Table 5.1 describe the simulation results of routing 100,000 virtual circuits into the network of Figure 5.9. In this network, the two paths have similar links but in different positions. Both paths receive on the average the same number of the virtual circuits and have the same average packet delay.

Although all the above five cases have the same total packet arrival rate, the average packet delay is different in each case with an extremely large average packet delay in case the last case ($\gamma = 1/100 \quad r = 1/7$), where each virtual circuit carries a large number of packets. This means that routing algorithms that consider only the packet arrival rate will achieve poor performance.

The more often that we update the link length information at the source node, the smaller average packet delay is achieved. The smaller the delay that the link length information becomes available to the source node, the smaller average packet delay is achieved. When the network state information is obsolete, the *Quadratic routing* seems to be slightly better than the *Shortest queue routing*, otherwise they achieve the same average packet delay.

The *Optimal static routing* assigns in a Round-Robin basis an odd numbered virtual circuit to path #1 and an even numbered virtual circuit to path #2.

When the updating period is not much larger than the mean interarrival time of virtual circuits, then both dynamic routing algorithms, *Quadratic routing* and *Shortest queue routing*, are clearly better than the *Optimal static routing*. However, when the updating period is extremely large compared to the mean interarrival time of virtual circuits, then the dynamic routing algorithms make many wrong decisions and therefore give larger average packet delay.

The *Shortest queue routing* is an approximation of the *Quadratic routing* and therefore they achieve similar average packet delay. Note also, that for single-link paths with equal link transmission speeds, both algorithms choose the same path. To see this, consider two single-link paths π and p, with link transmission speeds μ, N_π packets at path π link and N_p packets at path p link, such that

$$\frac{(1+N_\pi)^2}{\mu} < \frac{(1+N_p)^2}{\mu} \Leftrightarrow \frac{1+2*N_\pi+N_\pi^2}{\mu} < \frac{1+2*N_p+N_p^2}{\mu} \Leftrightarrow$$

$$\Leftrightarrow \frac{(N_\pi - N_p)*(N_\pi + N_\pi + N_p + 2)}{\mu} < 0 \Leftrightarrow \frac{N_\pi - N_p}{\mu} < 0 \Leftrightarrow$$

γ=1/7 r=1/100	instantaneous	obsolete
quadratic	19.02 ± 0.80	29.69 ± 1.06
shortest queue	18.77 ± 0.63	31.64 ± 1.27
optimal quasi-static	22.98 ± 1.83	

γ=1/14 r=1/50	instantaneous	obsolete
quadratic	14.19 ± 0.19	20.65 ± 0.85
shortest queue	13.97 ± 0.48	20.39 ± 0.78
optimal quasi-static	17.98 ± 0.62	

γ=1/26 r=1/27	instantaneous	obsolete
quadratic	15.24 ± 0.56	21.99 ± 0.63
shortest queue	15.43 ± 0.28	21.75 ± 0.58
optimal quasi-static	20.41 ± 0.57	

γ=1/50 r=1/14	instantaneous	obsolete
quadratic	24.47 ± 0.99	34.88 ± 1.47
shortest queue	23.38 ± 0.85	34.65 ± 1.17
optimal quasi-static	39.69 ± 1.47	

γ=1/100 r=1/7	instantaneous	obsolete
quadratic	53.88 ± 2.64	71.35 ± 0.82
shortest queue	53.72 ± 3.67	72.94 ± 2.26
optimal quasi-static	99.36 ± 5.89	

Table 5.1: The average packet delay ± error (95% confidence interval) for the network of Figure 5.9 with $\gamma = 1/7$, $r = 1/100$, for the *Quadratic* routing with instantaneous and obsolete information, the *Shortest queue* routing with instantaneous and obsolete information and the *Optimal quasi-static* routing implemented as Round-Robin.

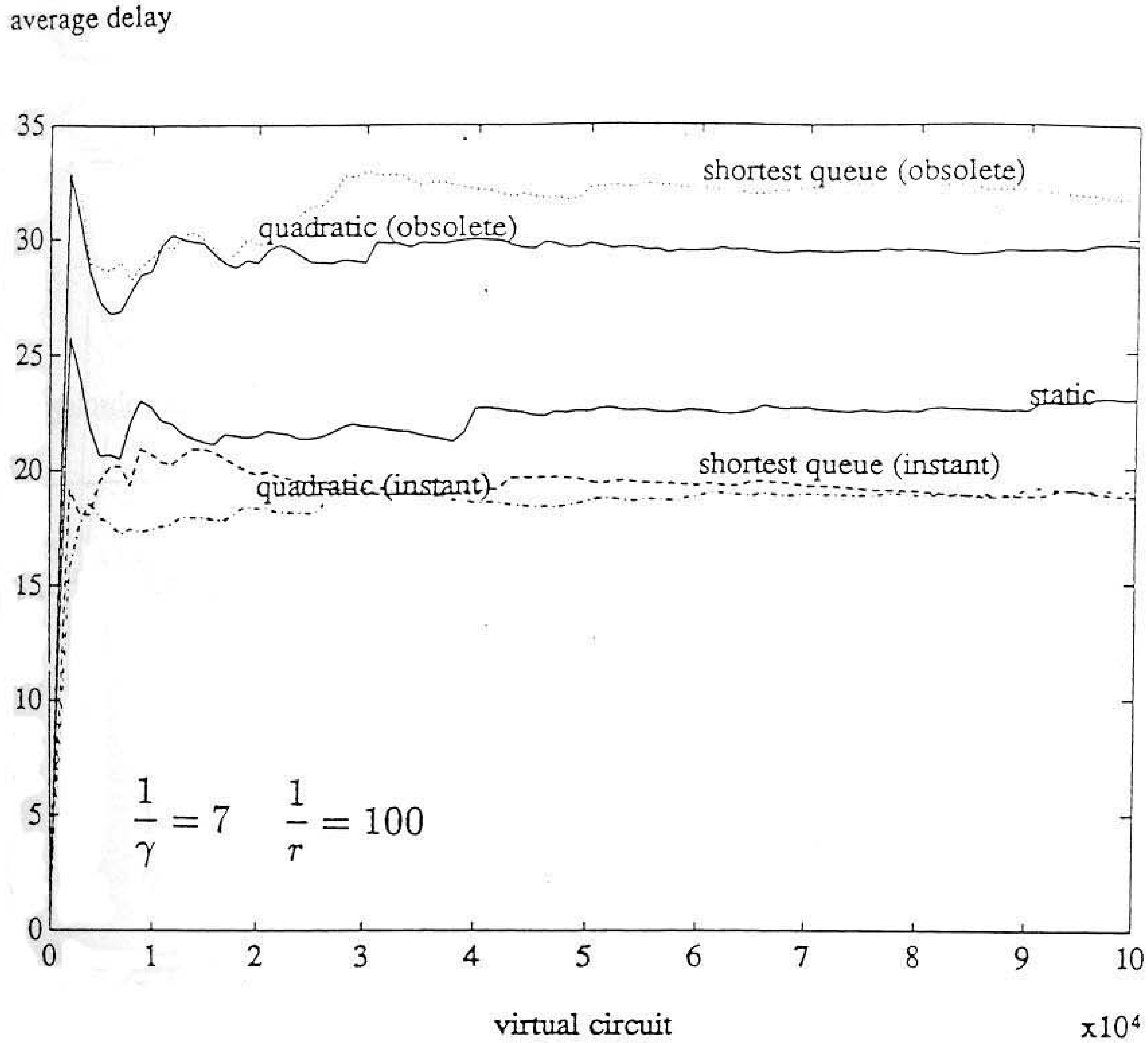

Figure 5.10: The average packet delay ± error (95% confidence interval) for the network of Figure 5.9 with $\gamma = 1/7$, $r = 1/100$, for the *Quadratic* routing with instantaneous and obsolete information, the *Shortest queue* routing with instantaneous and obsolete information and the *Optimal quasi-static* routing implemented as Round-Robin.

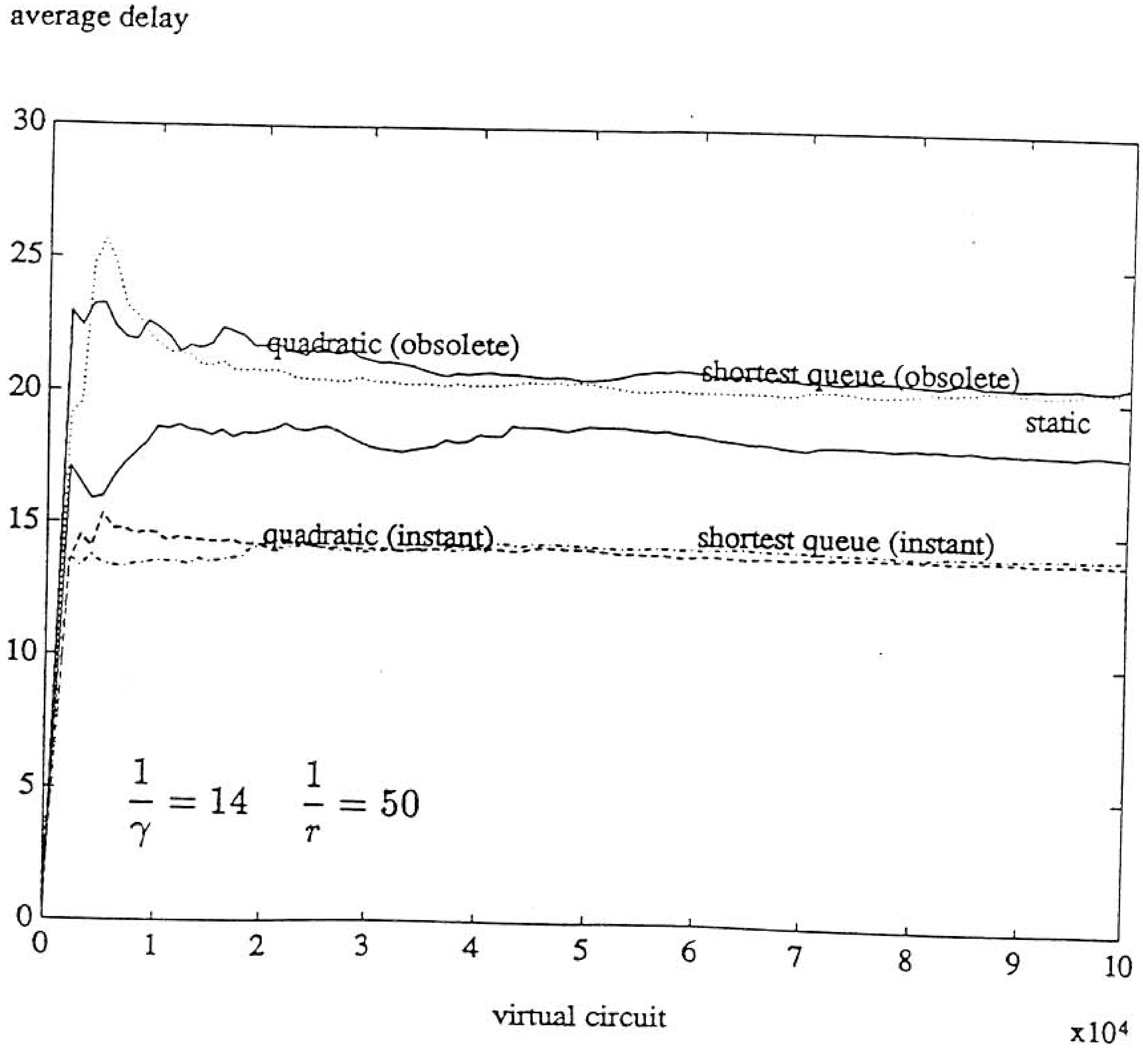

Figure 5.11: The average packet delay ± error (95% confidence interval) for the network of Figure 5.9 with $\gamma = 1/14$, $r = 1/50$, for the *Quadratic* routing with instantaneous and obsolete information, the *Shortest queue* routing with instantaneous and obsolete information and the *Optimal quasi-static* routing implemented as Round-Robin.

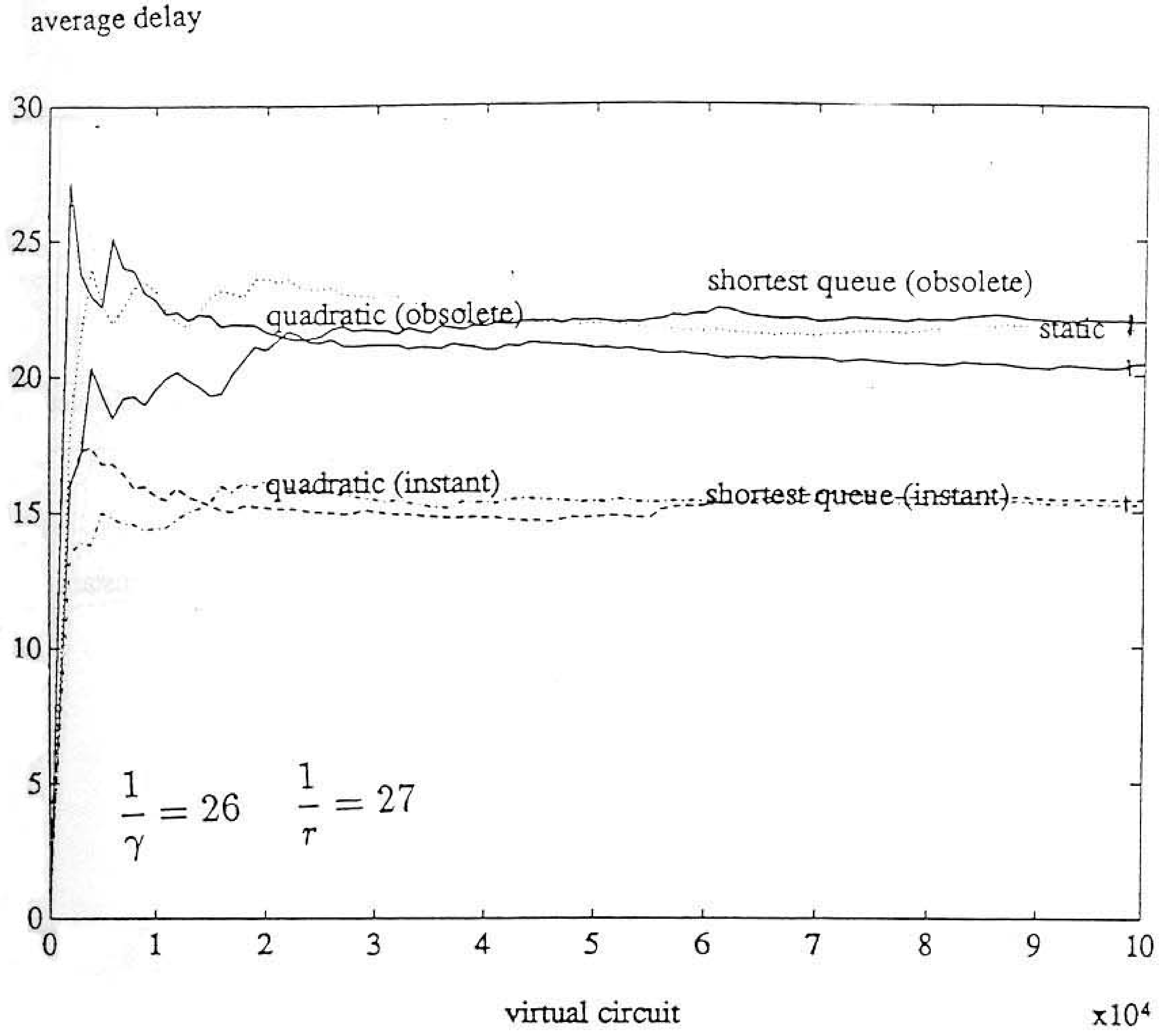

Figure 5.12: The average packet delay ± error (95% confidence interval) for the network of Figure 5.9 with $\gamma = 1/26$, $r = 1/27$, for the *Quadratic* routing with instantaneous and obsolete information, the *Shortest queue* routing with instantaneous and obsolete information and the *Optimal quasi-static* routing implemented as Round-Robin.

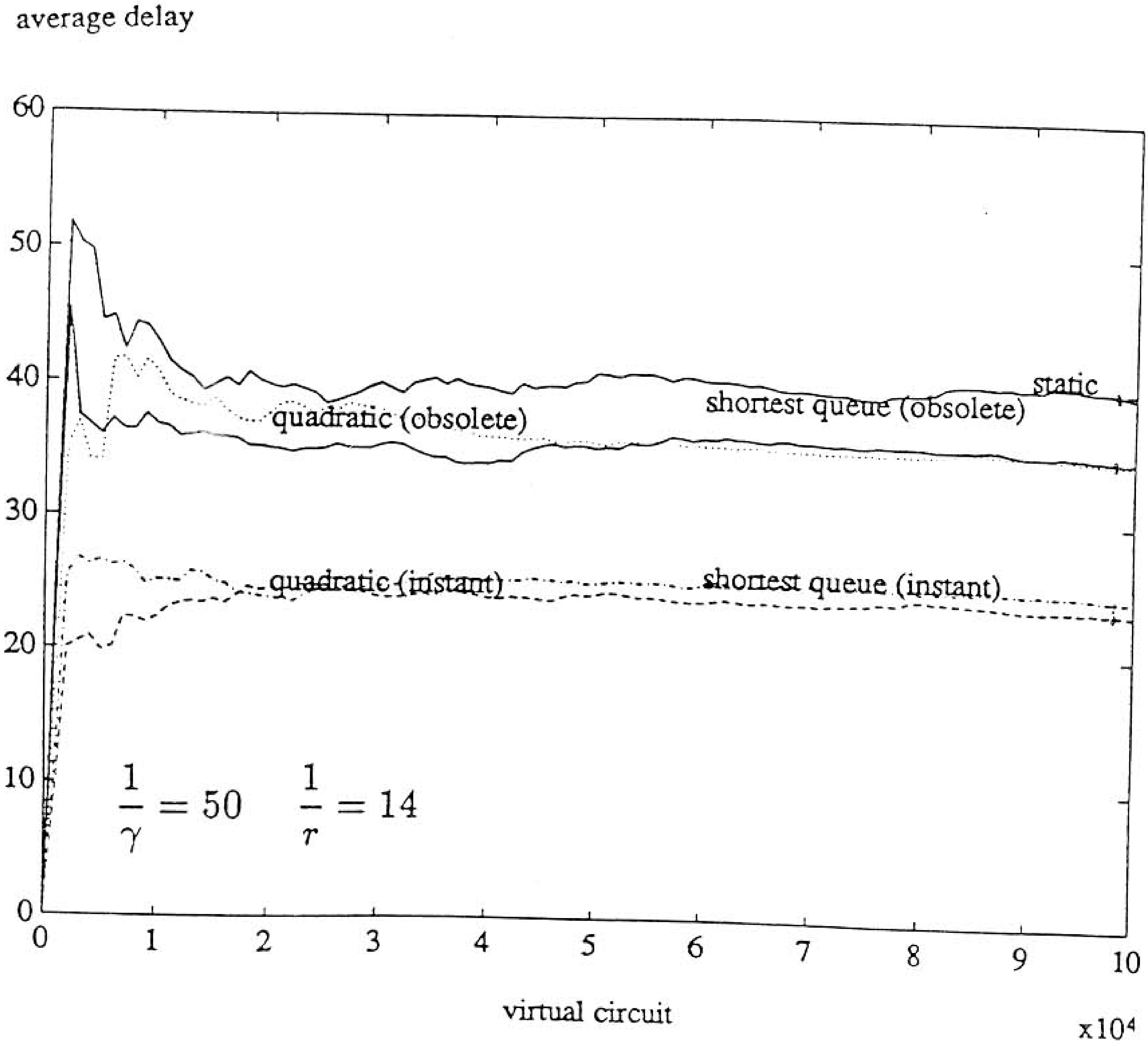

Figure 5.13: The average packet delay ± error (95% confidence interval) for the network of Figure 5.9 with $\gamma = 1/50$, $r = 1/14$, for the *Quadratic* routing with instantaneous and obsolete information, the *Shortest queue* routing with instantaneous and obsolete information and the *Optimal quasi-static* routing implemented as Round-Robin.

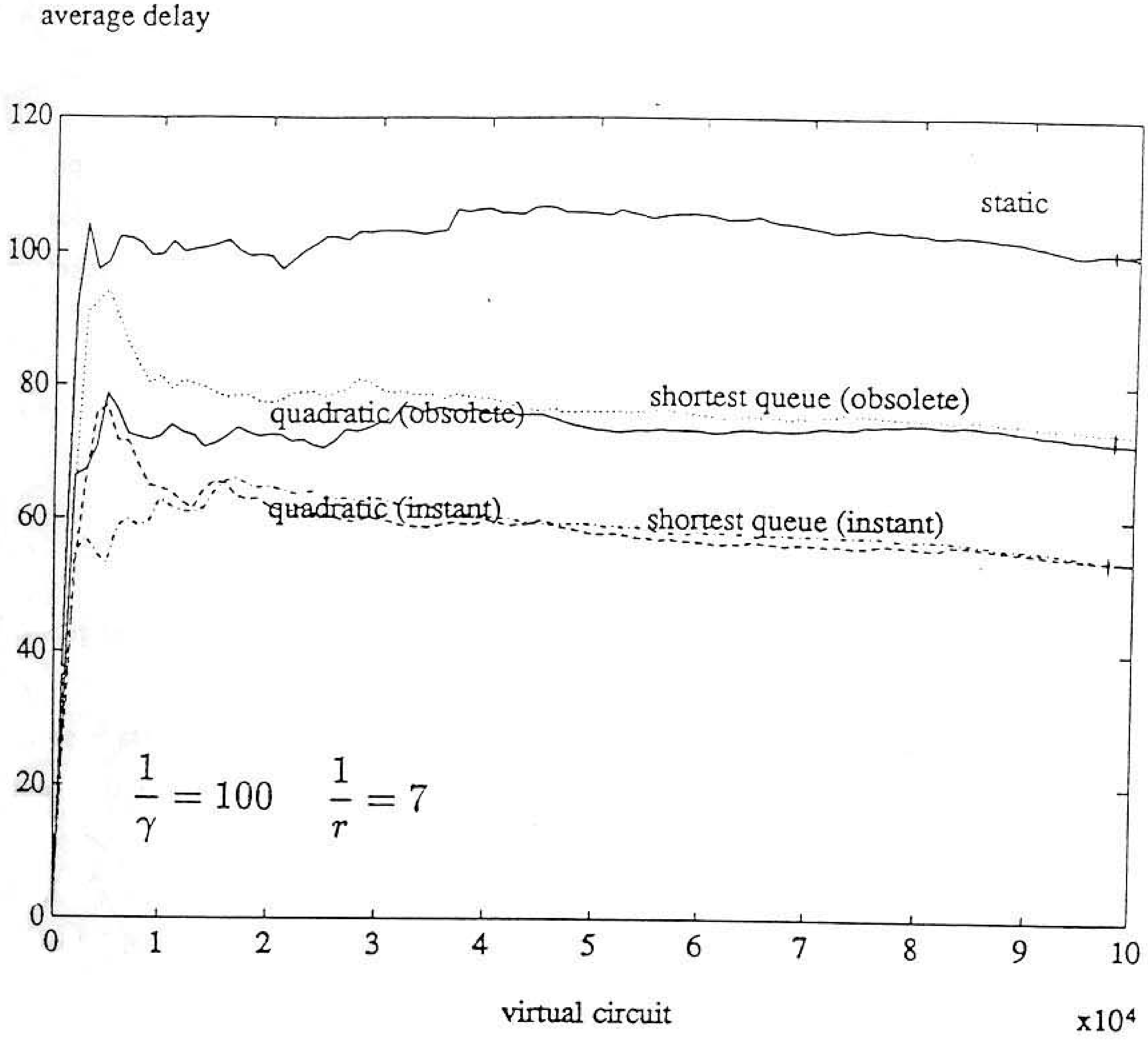

Figure 5.14: The average packet delay $\pm$ error (95% confidence interval) for the network of Figure 5.9 with $\gamma = 1/100$, $r = 1/7$, for the *Quadratic* routing with instantaneous and obsolete information, the *Shortest queue* routing with instantaneous and obsolete information and the *Optimal quasi-static* routing implemented as Round-Robin.

$$\Rightarrow \frac{N_\pi}{\mu} < \frac{N_p}{\mu} \Rightarrow \frac{1+N_\pi}{\mu} < \frac{1+N_p}{\mu}$$

That means that both algorithms choose path π since the ordering of the link lengths is the same for both algorithms.

In order that the *Quadratic routing* achieves different average packet delay than the *Shortest queue routing*, they should choose different paths for the same network state. Consider two paths π and p with the number of packets on their links satisfying the following relations simultaneously

$$\sum_{ij} \frac{(1+N_{ij})^2}{\mu_{ij}} * 1_{ij\in\pi} < \sum_{xy} \frac{(1+N_{xy})^2}{\mu_{xy}} * 1_{xy\in p}$$

$$\sum_{xy} \frac{1+N_{xy}}{\mu_{xy}} * 1_{xy\in p} < \sum_{ij} \frac{1+N_{ij}}{\mu_{ij}} * 1_{ij\in\pi}$$

then the *Quadratic routing* will choose path π, while the *Shortest queue routing* will choose path p.

Next, we further investigate the two dynamic algorithms for a more complex network with unbalanced paths. We consider a network with 5 paths from source to destination (Figure 5.15).

Path #1 has 3 links with transmission speeds 2, 1 and 3. Path #2 has 5 links with transmission speeds 4, 2, 0.5, 3 and 1. Path #3 has 7 links with transmission speeds 5, 1, 2, 3, 1, 4 and 2. Path #1 has 6 links with transmission speeds 1, 1, 1, 1, 1 and 1. Path #1 has 4 links with transmission speeds 2, 2, 2 and 2.

The mean packet service time is $\frac{1}{\mu} = 1$ and therefore $\mu_{ij} = \mu * C_{ij} = C_{ij}$. The mean virtual circuit duration is $\frac{1}{\delta} = 1000$. We consider two cases for the total packet arrival rate.

In case #1 The arrival rate of virtual circuits is $\gamma = \frac{1}{5}$ and the packet arrival rate per virtual circuit is $r = \frac{1}{50}$. Then the average number of virtual circuits into the network is $\frac{\gamma}{\delta} = 200$ and the average number of packets per virtual circuit is $\frac{r}{\delta} = 20$.

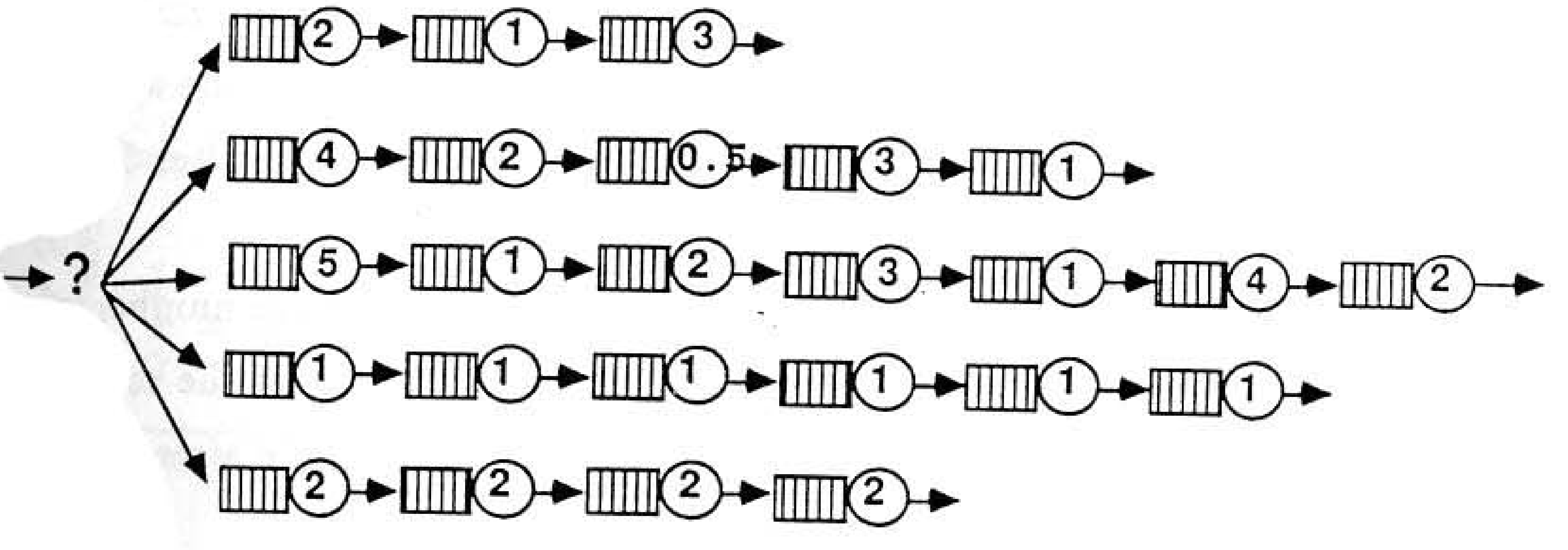

Figure 5.15: Simulated network.

In case #1 The arrival rate of virtual circuits is $\gamma = \frac{1}{50}$ and the packet arrival rate per virtual circuit is $r = \frac{1}{5}$. Then the average number of virtual circuits into the network is $\frac{\gamma}{\delta} = 20$ and the average number of packets per virtual circuit is $\frac{r}{\delta} = 200$.

The information at the source node about the link lengths in the network is updated according to four schemes:

a) *1 time unit*, when at every instant, the source node knows and uses the current number of packets at every link.

b) *20 time units*, when the information about the average number of packets at every link during a time interval of 20 time units is sent to the source node at the end of this time interval and it is used by the source node after 20 time units delay.

c) *50 time units*, when the information about the average number of packets at every link during a time interval of 50 time units is sent to the source node at the end of this time interval and it is used by the source node after 50 time units delay.

d) *100 time units*, when the information about the average number of packets at every link during a time interval of 100 time units is sent to the source node at the end of this time interval and it is used by the source node after 50 time units delay.

Figures 5.16, 5.17 and Table 5.2 describe the simulation results of routing 100,000 virtual circuits into the network of Figure 5.11. In this network, the paths are capacity inequivalent and they also have different number of links. Every path receives different number of virtual circuits and has different average packet delay. Similarly as in the previous network, the more often that we update the link length information at the source node, the smaller average packet delay is achieved. The smaller the delay that the link length information becomes available to the source node, the smaller average packet delay is achieved. However, the *Quadratic routing* achieves clearly smaller average packet delay than the *Shortest queue routing*, especially when the network state information becomes obsolete.

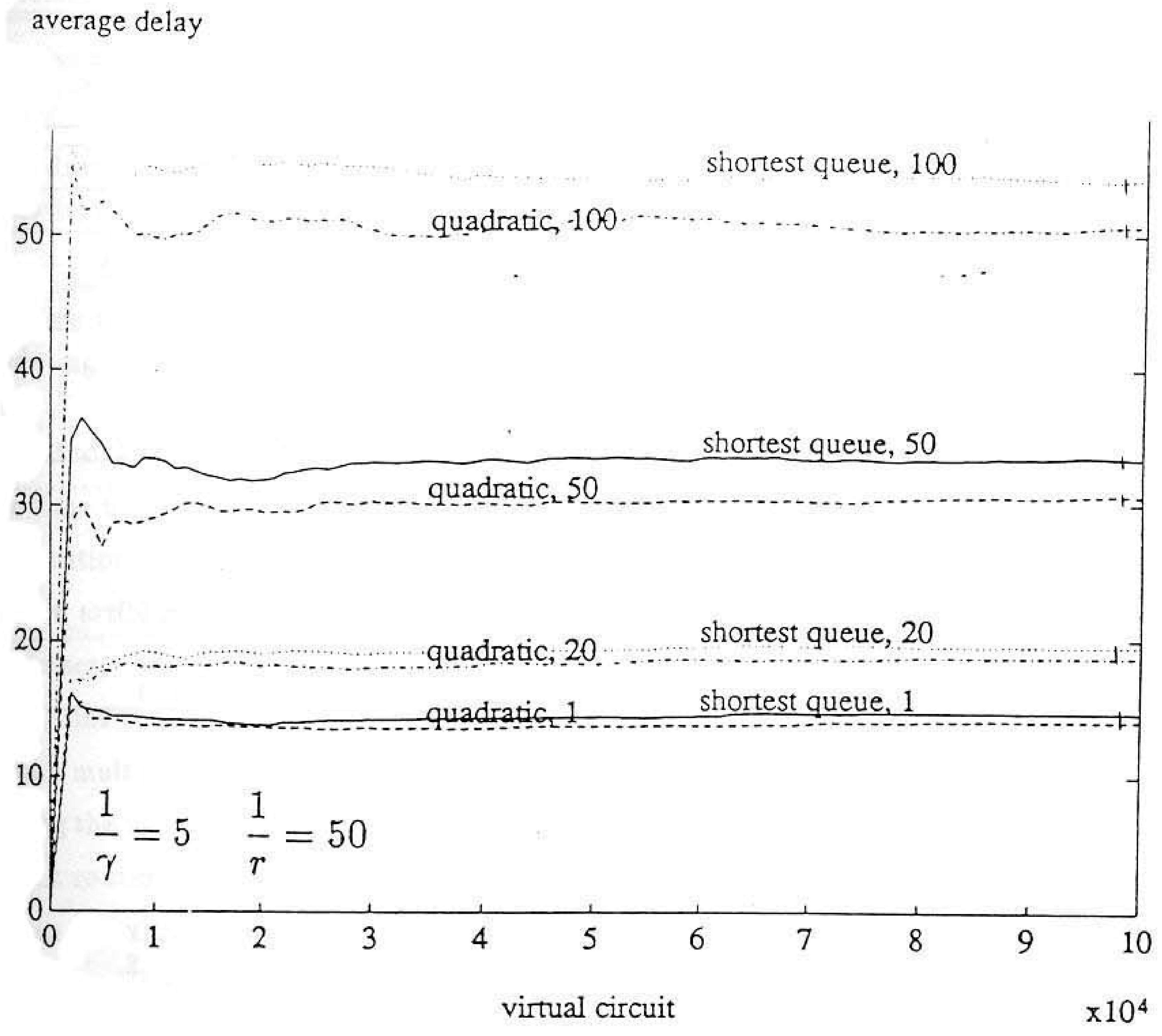

Figure 5.16: The average packet delay $\pm$ error (95% confidence interval) for the network of Figure 5.11 for $\gamma = 1/5$ $r = 1/50$, for the *Quadratic* and the *Shortest queue* routing with updating every 1, 20, 50, 100 time units.

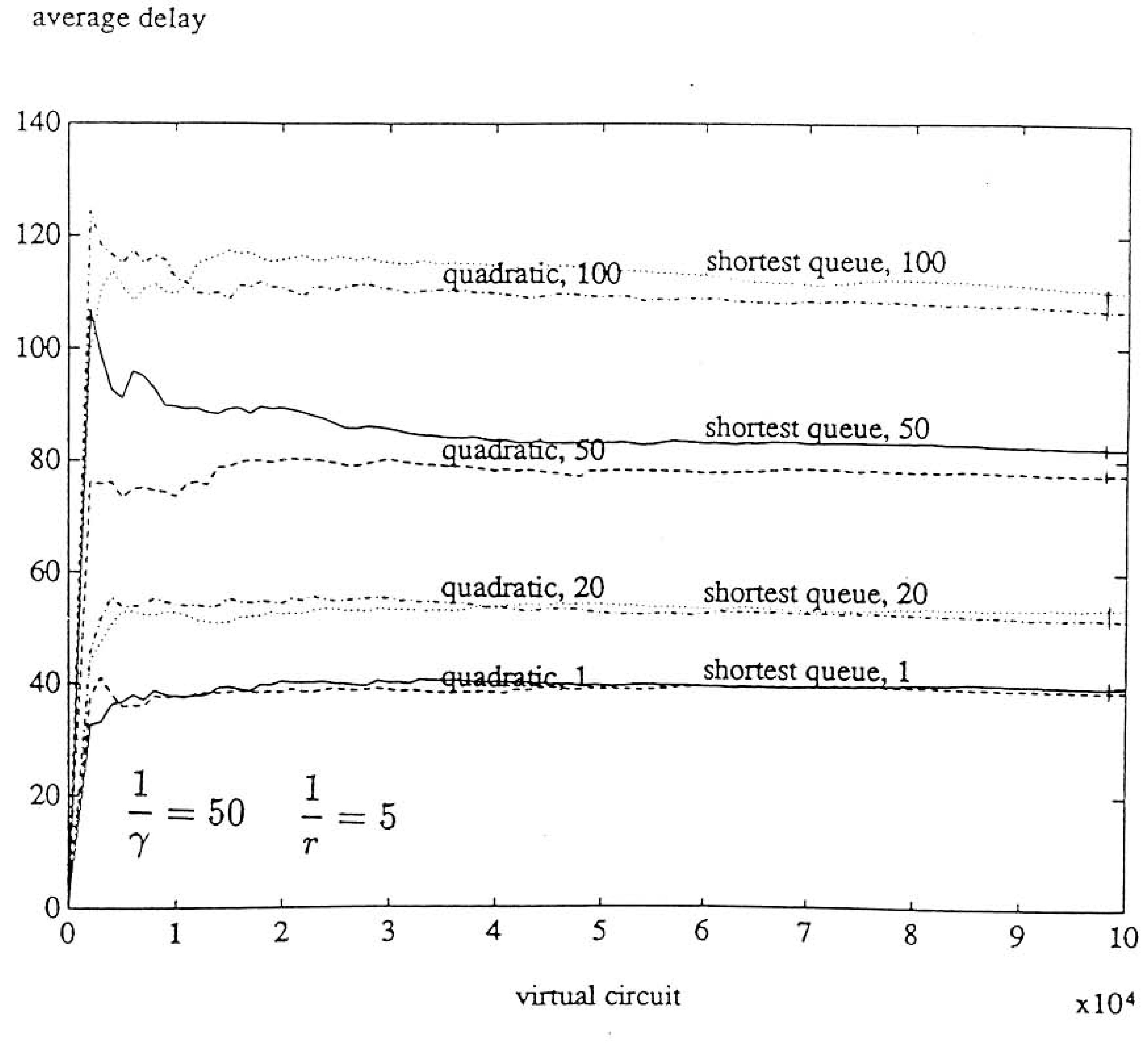

Figure 5.17: The average packet delay ± error (95% confidence interval) for the network of Figure 5.11 for $\gamma = 1/50$ $r = 1/5$, for the *Quadratic* and the *Shortest queue* routing with updating every 1, 20, 50, 100 time units.

$\gamma=1/5 \quad r=1/50$	1 time	20 time	50 time	100 time
quadratic	14.06 ± 0.27	18.74 ± 0.30	30.55 ± 0.54	50.70 ± 0.87
shortest queue	14.65 ± 0.25	19.51 ± 0.30	33.38 ± 0.42	54.13 ± 1.32

$\gamma=1/50 \quad r=1/5$	1 time	20 time	50 time	100 time
quadratic	38.98 ± 1.70	51.70 ± 1.84	77.53 ± 1.30	106.89 ± 1.61
shortest queue	39.59 ± 1.10	53.74 ± 0.81	82.21 ± 2.53	110.02 ± 2.62

Table 5.2: The average packet delay $\pm$ error (95% confidence interval) for the network of Figure 5.11 for the *Quadratic* and the *Shortest queue* routing with updating every 1, 20, 50, 100 time units.

Although for the above two cases, the total packet arrival rate is 4 packets per time unit, they give different average delay. This again confirm our previous observation that for virtual circuit networks is not enough to consider the aggregate packet arrival rate, but both the virtual circuit and packet per virtual circuit processes.

In this section, we present nonlinear dynamic queueing models of multi-destination multi-class virtual circuit networks, by explicitly considering the interaction among the virtual circuit and packet processes. We formulate the integrated virtual circuit routing and congestion control problem as an optimal control problem. We set up a multi-objective function and we solve it using the Pontryagin maximum principle. Then we derive state dependent routing and congestion control policies for virtual circuit network control and we define as link length a quadratic function of the average number of packets on it. Finally, we demonstrate via simulation, that for an unbalanced network, this Quadratic routing achieves smaller average packet delay than a Shortest queue routing. For a balanced network, both the Quadratic routing and the Shortest queue routing achieve similar average packet delay, that is also smaller than that achieved by the Optimal quasi-static routing, when the updating period is not extremely larger than the mean interarrival time of virtual circuits.

5.7 Application to Integrated Services Networks

In this section, we apply the methodologies developed in the previous sections to integrated services networks. In section 6, for each class, we model the traffic processes in two interacting levels: i) the virtual circuit process level and ii) the packet process level. For integrated services networks, we propose using more than two interacting levels. For example, four levels: i) subscriber level, ii) virtual circuit level, iii) burst level, iv) packet level.

Different dynamic queueing models (such as those of section 5.4 for datagram networks) will be used at each level to model the dynamic evolution of the corresponding processes.

Furthermore, we can also introduce other dynamic models based on finite population queueing models. For example, at the subscriber level, let $A(t)$ be the active number of subscribers among the existing $S(t)$, $a(t)$ the rate at which an idle subscriber becomes active and $b(t)$ the rate at which an active subscriber becomes idle. Then a dynamic queueing model that describes the average number of active subscribers is the following:

$\dot{A}(t) = a(t) * (S(t) - A(t)) - b(t) * A(t)$

Now, each active subscriber creates virtual circuits, and each virtual circuit creates bursts, and each burst creates packets as in section 5.6. Therefore the state of each system resource is described by three variables: the number of active subscribers $A(t)$, the number of virtual circuits $V(t)$, the number of bursts $B(t)$ and the number of packets $N(t)$ at this resource. So, for each class c, the state of a resource is

$$\mathbf{X}^c(t) = [A^c(t), V^c(t), B^c(t), N^c(t)]$$

Similarly, for the cost functions, we add to the costs of section 5.6 another level of costs for the active number of subscribers, bursts etc.

Chapter 6

Stochastic Learning Automata for Decentralized Load Sharing, Routing & Congestion Control

In this chapter, we introduce another novel methodology for decentralized dynamic load sharing, routing and congestion control. We propose stochastic learning automata at the source nodes of the system for admitting or rejecting jobs, for selecting the destination node for job processing, and for selecting the routing path to the destination node. These decisions will be done probabilistically by learning automata algorithms that will update their action probabilities according to measurements of the path and source-destination lengths. The path and source-destination lengths are those derived in the dynamic optimality conditions of chapter 4. We also introduce novel classes of stochastic learning automata:
i) state dependent learning automata, whose adaptation rates are functions of the system state, ii) two-step learning automata, that use larger adaptation rates when the selected action repeatedly gives good performance, iii) multiple response automata, that use different adaptation rates for different system learning response (not just the favorable/unfavorable response of previous learning automata). We prove that these learning automata are feasible at each step, non-absorbing, strictly distance diminishing, ergodic and expedient. We apply this methodology to datagram, virtual circuits and integrated services networks. We give an example, where we make virtual circuit routing decisions learning automata algorithms. We show (via simulation) that by suitable tuning the adaptation rates of the algorithms, the learning automata achieve smaller average packet delay. We also show that

a path length proposed in chapter 5, is superior to a shortest-queue-type routing, usually used in real networks.

6.1 Introduction

Learning is defined as any relatively permanent change in behavior resulting from past experience, and a learning system is characterized by its ability to improve its behavior with time, in some sense tending towards an ultimate goal. In mathematical psychology, learning systems [80, 15, 494, 259] have been developed to explain behavior patterns among living organisms. These mathematical models in turn have lately been adapted to synthesize engineering systems [344].

Tsetlin [495] initially introduced the concept of learning automaton operating in an unknown random environment. He considered learning behaviors of finite deterministic automata under the stationary random environment. Varshavskii & Vorontsova [503] introduced variable structure stochastic automata in an unknown random environment. Chandrasekaran & Shen [92] Poznyak [386], Tsypkin & Poznyak [498], Flerov [163], Polyak [383, 384] Lakshmivarahan & Thathachar [283] and others further advanced the learning automaton theory.

A number of books on learning automaton theory have been also appeared. Norman [355, 356, 357] develops a Markov process-based approach to analyze the learning automaton and explain the learning processes in organisms. Lakshmivarahan [282] provides a rigorous analysis of the learning automaton theory. El-Fattah [143] presents learning automata used for pattern recognition systems and for simulation of collective behavior problems. Glorioso & Osorio [197] describe fundamental issues of learning and applications in engineering. Baba [19] presents learning automaton behavior under unknown multi-teacher environments. Narendra & Thathachar [342] provide a rigorous introduction to the theory of learning automata.

A number of papers have been also appeared recently [13, 281, 482, 367, 364, 363, 365, 451] that propose new reinforcement schemes for learning automata and investigate their properties.

Learning automata have been also applied to pattern recognition problems [22], to routing problems [346, 343, 456], to flow control problems [321, 322], to partitioning problems [366], to neural network models [24, 23, 473] etc.

In the previous two chapters, we found the conditions for team optimality, Nash and Stackelberg equilibrium for the joint load sharing, routing and congestion control problem. In this chapter, we introduce stochastic learning automata as decentralized decision-makers that will achieve these conditions. A stochastic learning automaton is an adaptive control algorithm that reacts to the system's response. It chooses an action and if the system's response is favorable, then it reinforces that action, otherwise it tends to choose another action.

The greatest potential of the learning automata methodology is that it permits the analysis of very complex dynamic systems, and global optimization is possible. Even when little information is available, they tend to stabilize a nonstationary system by predicting its behavior.

We propose using learning automata at the source nodes of the system for admitting or rejecting a job from the system (congestion control), for selecting the destination computer site for processing the job (load sharing) and selecting the path through which the job will reach this destination node (routing).

In previous chapters, we found what the optimal load sharing, routing and congestion control policies should be. These optimal control policies may be implemented directly as they were found. However, the underlying assumptions of the models (e.g. independent exponential distributions), or even other management problems that were not explicitly considered, may affect these optimal control policies. Therefore, we propose the use of learning automata that will drive to these optimal control policies. So, instead of deterministically choosing the minumum length path, learning automata will choose it with very high probability. Note, that if we appropriatelly calibrate the step size of these learning automata algorithms, then they may choose the minimum length path with probability 1.

6.2 Learning Automaton Theory

In this section, we review the basic learning automaton theory [344, 342].

A learning automaton is a feedback system (Figure 6.1) connecting a stochastic automaton $(X, \phi, a, \mathbf{P}, T, G)$ and an environment $C = \{C_1, \ldots, C_{|a|}\}$, where
X: input set or environment response.

1) if $X \in \{0,1\}$, i.e. the environment response takes only two values, where $X = 0$ can be considered as reward and $X = 1$ as penalty, we have a P-model

2) if $X \in \{X^1, \ldots, X^k\}$, $X^i \in [0,1]$, i.e. the environment response takes a certain number of values in the interval [0,1], with $X = 0$ to be "full reward" and $X = 1$ to be "full penalty", we have a Q-model.

3) if $X \in [0,1]$, i.e. the environment response takes any value in the interval [0,1], with $X = 0$ to be "full reward" and $X = 1$ to be "full penalty", we have an S-model.

$\phi = \{\phi_1, \ldots, \phi_s\}$, $s < \infty$: set of internal states.
$a = \{a_1, \ldots, a_{|a|}\}$, $|a| \leq s$: output or action set.
$\mathbf{P}(n) = [P_1(n), \ldots, P_{|a|}(n)]^T$ state probability vector, where $P_i(n) = P[a(n) = a_i]$.
T: algorithm, updating or reinforcement scheme, that generates the action probability $\mathbf{P}(n+1) = T[\mathbf{P}(n), a(n), X(n)]$.
G: $\phi \to a$: output function.
$c_i(n) = E[X(n)/a(n) = a_i]$: expected penalty for action a_i, which are unknown and there is a unique minimum.

In order to evaluate a learning automaton, some measures are defined [344]:

1) **Expedience** : if at a certain time instant n, the automaton selects action a_i with probability $P_i(n)$, then the average penalty received by the automaton conditioned on $\mathbf{P}(n)$ is $M(n) = E[X(n)/\mathbf{P}(n)]$. If no a priori information is available and the actions are chosen at random then $M_0 = (C_1 + \ldots + C_{|a|})/|a|$. A learning automaton is called expedient if $\lim_{n\to\infty} E[M(n)] < M_0$ and of course it performs better than one that selects its actions randomly.

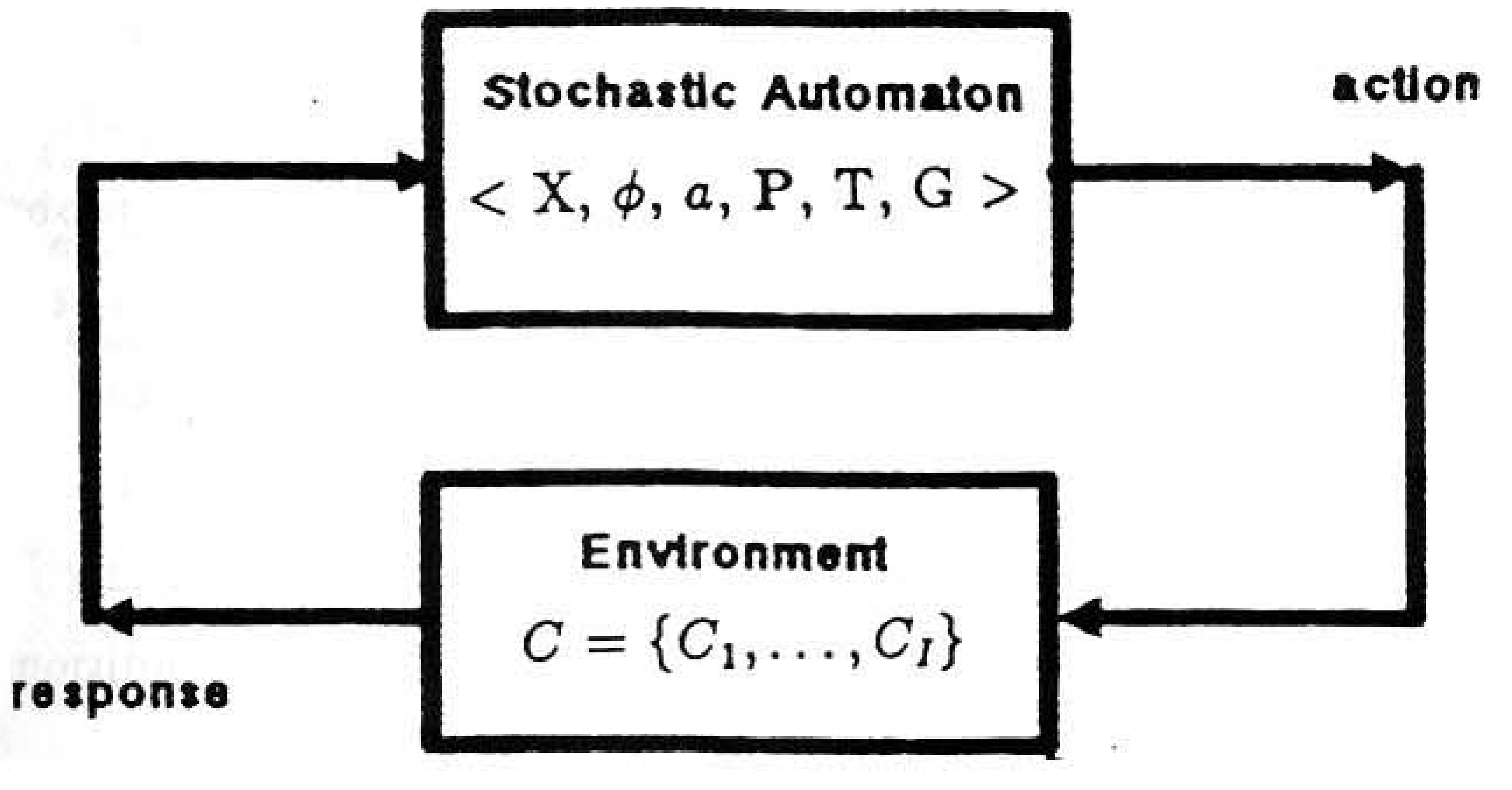

Figure 6.1: Learning Automaton.

2) **Optimality** : a learning automaton is called optimal if

$$\lim_{n\to\infty} E[M(n)] = c_l, \quad \text{where} \quad c_l = min\{c_1, \ldots, c_{|a|}\}$$

or if

$$\lim_{n\to\infty} E[P_l(n)] = 1, \quad \text{where} \quad P_l(n) = P[a(n) = a_l]$$

i.e. optimality means that asymptotically the action associated with the minimum expected penalty is chosen with probability one.

A learning automaton is called ϵ-optimal if

$$\lim_{n\to\infty} E[M(n)] < c_l + \epsilon, \ \epsilon > 0$$

Next, the operation of a learning automaton is described: The automaton selects action $a(n) = a_i$ with probability $P_i(n)$ at each instant n. Action $a(n)$ becomes input to the environment (Figure 6.1). If this results in a favorable outcome for the network performance ($X(n) \to 0$), then the probability $P_i(n)$ is increased by $\Delta P_i(n) = P_i(n+1) - P_i(n)$ and the $P_j(n), j \neq i$, are decreased by $\Delta P_j(n) = P_j(n+1) - P_j(n)$. Otherwise, if an unfavorable outcome ($X(n) \to 1$) appears, then the $P_i(n)$ is decreased by $\Delta P_i(n) = P_i(n+1) - P_i(n)$ and the $P_j(n), j \neq i$ are increased by $\Delta P_j(n) = P_j(n+1) - P_j(n)$. By iteration of the algorithm, we achieve adaptation to varying environment conditions.

Let $a(n) = a_i$ and
if $X(n) \to 0$ then

$$P_i(n+1) = P_i(n) + \sum_{j\neq i} f_j[\mathbf{P}(n)]$$

$$P_j(n+1) = P_j(n) - f_j[\mathbf{P}(n)] \ , \forall j \neq i$$

else $X(n) \to 1$ then

$$P_i(n+1) = P_i(n) - \sum_{j\neq i} g_j[\mathbf{P}(n)]$$

$$P_j(n+1) = P_j(n) + g_j[\mathbf{P}(n)] \ , \forall j \neq i$$

where f_j and g_j are nonnegative continuous functions and $0 < P_i(n) < 1$, $\sum_{i=1}^{|a|} P_i(n) = 1$. f_j and g_j can be linear or nonlinear functions of $\mathbf{P}$ (n). A class of linear algorithms is :

$$f_j(\mathbf{P}(n)) = \alpha * P_j(n), \quad 0 < \alpha < 1, \ \alpha : \text{reward parameter}$$

$$g_j(\mathbf{P}(n)) = \beta * \left[\frac{1}{|a| - 1} - P_j(n)\right], \quad 0 \leq \beta < 1, \ \beta : \text{penalty parameter}$$

Three linear schemes exhibit interesting behavior [344]. In the L_{R-I} (Linear Reward-Inaction) algorithm ($\beta = 0$), every sample path converges to selecting only one action with probability one and it is ϵ-optimal. For the L_{R-P} (Linear Reward Penalty) algorithm ($\beta = \alpha$), and the $L_{R-\epsilon P}$ (Linear Reward Infinitesimal Penalty) algorithm ($\beta << \alpha$), $\mathbf{P}(n)$ converges in distribution to a random vector $\mathbf{P}$, whose distribution is independent of $\mathbf{P}(0)$. Further the $L_{R-\epsilon P}$ is ϵ-optimal and not be locked in on a nonoptimal action.

For the $L_{R-\epsilon P}$ algorithm [343, 456] there is a unique $\mathbf{P}^*$ such that $C_i(\mathbf{P}^*) = C_j(\mathbf{P}^*)$ $\forall\, i, j$, i.e. in the limit the expected penalties of the actions are equalized and the action corresponding to the lowest expected penalty is chosen with probability close to 1.

6.3 State Dependent Learning Automata

In this section, we extend the learning automata theory by making the updating scheme a function of the environment state.

At time n, action i is selected according to an action probability $P_i(n) \geq 0$, $\sum_{i=1}^{A} P_i(n) = 1$. We define as response $X_i(n)$ of the environment a continuous, monotonous, non-decreasing function of the cost $C_i(n)$ of the selected action i normalized to the [0,1] interval, i.e.

$$X_i(n) = \varphi(C_i(n)) \qquad 0 \leq X(n) \leq 1$$

In this way, we correspond $X_i(n) \rightarrow 0$ to a favorable outcome (small $C_i(n)$), and $X_i(n) \rightarrow 1$ to an unfavorable outcome (large $C_i(n)$). Examples of such functions are :

$$X_i(n) = \frac{C_i(n)}{C_{max}(n)}$$

$$X_i(n) = \frac{C_i(n) - C_{min}(n)}{C_{max}(n) - C_{min}(n)}$$

$$X_i(n) = e^{\alpha * (C_i(n) - C_{max}(n))}$$

The choice of the suitable function depends on our desire to stress some response areas or to have uniform adaptation speed to the response.

If the environment's response $X_i(n)$ to the selected action i is the minimum among all alternative actions, then its action probability increases by $\Delta P_i(n)$ and the action probabilities of the other actions are decreased. Otherwise, its action probability is decreased by $\Delta P_i(n)$ and the action probabilities of the other actions are increased.

Then we propose the following *State-Dependent (SD) algorithm*:

Let $\quad a(n) = a_i$

If $\quad X_i(n) = \min_j \{X_j(n)\}$, then

$$P_i(n) = P_i(n-1) + \alpha * [1 - X(n)] * \sum_{j \neq i} f_j[\mathbf{P}(n)]$$

$$P_j(n) = P_j(n-1) - \alpha * [1 - X(n)] * f_j[\mathbf{P}(n)]$$
$$\forall j \neq i$$

else

$$P_i(n) = P_i(n-1) - \beta * X(n) * \sum_{j \neq i} g_j[\mathbf{P}(n)]$$

$$P_j(n) = P_j(n-1) + \beta * X(n) * g_j[\mathbf{P}(n)]$$
$$\forall j \neq i$$

where $\quad 0 < \alpha, \beta < 1$.

where f_j and g_j are nonnegative continuous functions and $0 < P_i(n) < 1$, $\sum_{i=1}^{|a|} P_i(n) = 1$. f_j and g_j can be linear or nonlinear functions of $\mathbf{P}(n)$.

If the reward and penalty functions are linear functions of the action probabilities, then we have a *State-Dependent Linear (SDL) algorithm*.

6.4 Two-Step Learning Automata

In this section, we propose a learning automaton that uses two action levels to update its action probabilities. If the action that is chosen at step n is the best, and was also the best at step $n-1$, then we reward this action a lot by increasing its probability with a large step size. If the action that is chosen at step n is the best, but was not the best in the previous step, then we reward this action a little by increasing its probability with a small step size. Otherwise, if the action that is chosen at step n is not the best, and was also not the best at step $n-1$, then we penalize this action a lot by decreasing its probability with a large step size. If the action that is chosen at step n is not the best, but was the best in the previous step, then we penalize this action a little by decreasing its probability with a small step size. The above concepts leads us to the following *Two-Step algorithm*:

Let $a(n) = a_i$:

$$If \quad X_i(n) \quad = \min_j\{X_j(n)\}, then$$

$$if \qquad X_i(n-1) = \min_j\{X_j(n-1)\}, then$$

$$P_i(n) = P_i(n-1) + \alpha^1 * [1 - P_i(n-1)]$$

$$P_j(n) = P_j(n-1) - \alpha^1 * P_j(n-1) \quad \forall j \neq p$$

$else$

$$P_i(n) = P_i(n-1) + \alpha^2 * [1 - P_i(n-1)]$$

$$P_j(n) = P_j(n-1) - \alpha^2 * P_j(n-1) \quad \forall j \neq p$$

else

$$if \quad X_i(n-1) = \min_j\{X_j(n-1)\}, then$$

$$P_i(n) = P_i(n-1) - \beta^2 * P_i(n-1)$$

$$P_j(n) = P_j(n-1) + \beta^2 * \left[\frac{1}{|a|-1} - P_j(n-1)\right] \ \forall j \neq p$$

else

$$P_i(n) = P_i(n-1) - \beta^1 * P_i(n-1)$$

$$P_j(n) = P_j(n-1) + \beta^1 * \left[\frac{1}{|a|-1} - P_j(n-1)\right] \ \forall j \neq p$$

where $0 < \alpha_2 < \alpha_1, \quad \beta_2 < \beta_1 < 1.$

6.5 Virtual Updating

Another way to update the action probabilities with less overhead, (but also less accuracy) is to update less frequently, for example at times τ_n. We consider two cases according to if we observe or not the environment between the update instants:

6.5.1 Observable State

We assume that action selection (based on the $P_i(t_k)$'s) is made at the update points. Knowing that there are $n_i(\tau_n)$ successes (favorable outcomes) and $u_i(\tau_n)$ failures (unfavorable outcomes) during $[\tau_n, \tau_{n+1})$, then we must increase the action probability of the selected action $n_i(\tau_n)$ times and decrease it $u_i(\tau_n)$ times. In a similar way we must decrease and increase the action probabilities of the other actions.

Since we do not want to keep track of the exact sequence of occurrence of failures and successes, we assume such sequences. There are several ways to accomplish this, for example :

i) Increase P_i in $n_i(\tau_n)$ updates, then decrease it in $u_i(\tau_n)$ updates.

ii) Let $n_i(\tau_n) \leq u_i(\tau_n)$. Increase and decrease P_i in $n_i(\tau_n)$ updates, then decrease it in $u_i(\tau_n) - n_i(\tau_n)$ updates.

iii) Let $n_i(\tau_n) > u_i(\tau_n)$. Increase and decrease P_i in $u_i(\tau_n)$ updates, then increase it in $n_i(\tau_n) - u_i(\tau_n)$ updates.

iv) Decrease P_i in $u_i(\tau_n)$ updates, then increase it in $n_i(\tau_n)$ updates.

v) Let $n_i(\tau_n) \leq u_i(\tau_n)$. Decrease and increase P_i in $n_i(\tau_n)$ updates, then decrease it in $u_i(\tau_n) - n_i(\tau_n)$ updates.

vi) Let $n_i(\tau_n) > u_i(\tau_n)$. Decrease and increase P_i in $u_i(\tau_n)$ updates, then increase it in $n_i(\tau_n) - u_i(\tau_n)$ updates.

We can solve these recurrence equations and have $P_i(\tau_{n+1}) = Function\ (P_i(\tau_n), n_i(\tau_n), u_i(\tau_n))$. Thus instead of updating $P_i(\tau_n)$ at every action success or failure, we update at the times τ_n.

A simpler idea is to weight the reward adaptation step size with the number of successes and the penalty step size with the number of failures. So, instead of α, we can use

$$\alpha * \frac{n_i(\tau_n)}{n_i(\tau_n) + u_i(\tau_n)}$$

and instead of β, we can use

$$\beta * \frac{u_i(\tau_n)}{n_i(\tau_n) + u_i(\tau_n)}$$

6.5.2 Non Observable State

Another way to update the action probabilities multiple times is based on a single measurement. If the system state does not change too rapidly, then we assume that the same outcome would have been repeated if we were continually measuring the system state, say l times until the next real measurement. So, we update the probabilities l times assuming the last outcome still holds. Note that the updating scheme is composed of recursive equations. This leads us to extend the previously proposed updating scheme by using one network state measurement, but many (for example l_k in region k) iterations of the scheme in one actual computing step (updating step from *n-1* to n). For clarity we show the transformation of only one network response region (the full detail is given in the appendix).

If $X_i(n) \leq \phi_1(m(n))$, then

$$\begin{aligned} P_i(n) &= P_i(n-1) * \{1 - \alpha_1 * [1 - X(n)]\} + \\ &+ \alpha_1 * [1 - X(n)] \end{aligned}$$

$$P_j(n) = P_j(n-1) * \{1 - \alpha_1 * [1 - X(n)]\} \quad \forall j \neq i$$

Since the measurements for $X_j(n)$ do not change between $n-1$ and n, then $P_j(n)$ did not not change according to the previous updating schemes, so call them X_j and P_j. However, we shall update the action probabilities P_j multiple times, say l_1, based on the measurements X_j. So, by solving these recursive equations, we have the following equations

$$\begin{aligned}
P_i(l_1) &= P_i * \{1 - \alpha_1 * [1 - X]\}^{l_1} + \\
&+ \alpha_1 * [1 - X] * \sum_{i=0}^{l_1 - 1} \{1 - \alpha_1 * [1 - X]\}^i
\end{aligned}$$

$$P_j(l_1) = P_j * \{1 - \alpha_1 * [1 - X]\}^{l_1} \quad \forall j \neq i$$

and the updating scheme becomes

If $X_i(n) \leq \phi_1(m(n))$, then

$$\begin{aligned}
P_i(n) &= P_i(n-1) * \{1 - \alpha_1 * [1 - X(n)]\}^{l_1} + \\
&+ \alpha_1 * [1 - X(n)] * \sum_{i=0}^{l_1 - 1} \{1 - \alpha_1 * [1 - X(n)]\}^i
\end{aligned}$$

$$P_j(n) = P_j(n-1) * \{1 - \alpha_1 * [1 - X(n)]\}^{l_1} \quad \forall j \neq i$$

We can use different $l_k, k = 1, ..., R$ and $m_k, k = 1, ..., P$ for different regions, where $l_1 \geq l_2 \geq ... \geq l_R > 0$, and $m_1 \geq m_2 \geq ... \geq m_P > 0$, are positive integers.

6.5.3 Frequent Updating

In the previous section, we updated as little as possible in order to reduce the measurement and computation overhead. However, the best results will be achieved if we measure and update the action probabilities as often as possible. Then the action algorithm will track the system state faster and the decisions will be better. Of course this will introduce more overhead of transmitting, selecting, storing and computing the state statistics.

6.6 Multiple Response Learning Automata

In this section, we introduce Multiple Response (MR) learning automata algorithms. The idea is to use different adaptation rates for different environment responses ($X(n)$). If the environment response is far away from optimum, the algorithm should converge faster, while if the environment response is near to optimum the algorithm should have smaller fluctuation. Whenever the environment response is very good ($X(n) \to 0$) (reward response 1), then the probability of the selected action increases very fast ($\alpha \to 1$). When the environment response is almost good (reward response R), then the probability of the selected action increases slowly ($\alpha \to 0$). Correspondingly, whenever the environment response is very bad ($X(n) \to 1$) (penalty response 1), then the probability of the selected action decreases very fast ($\beta \to 1$). When the cost of the environment response is almost bad (penalty response P), then the probability of the selected action decreases slowly ($\beta \to 0$).

6.6.1 Q-MR Learning Automata

In this section, we introduce a Q-model MR learning automaton algorithm, for which the environment's response takes discrete values. So, if action a_i was selected at time n, the environment's response is an element of the set
$\{X_i^1, \ldots, X_i^R, \bar{X}_i^P, \ldots, \bar{X}_i^1\}$, i.e.

Let $a(n) = a_i$

reward response 1: $X(n) = X_i^1(X(n))$

reward response 2: $X(n) = X_i^2(X(n))$

...

reward response R: $X(n) = X_i^R(X(n))$

penalty response P: $X(n) = \bar{X}_i^P(X(n))$

penalty response P-1: $X(n) = \bar{X}_i^{P-1}(X(n))$

...

penalty response 1: $X_i(n) = \bar{X}_i^1(X(n))$

where $0 \leq X_i^1 < X_i^2 < ... < X_i^R < m_i < \bar{X}_i^P < \bar{X}_i^{P-1} < ... < \bar{X}_i^1 \leq 1$ are functions of $X(n)$.

A possible sequence for these functions $\{X_i^r(X(n))\}$ could be a Fibonacci sequence (normalized to the $[0, m_i(X(n))$ interval). Also a possible sequence for the functions $\{\bar{X}_i^p(X(n))\}$ could be a Fibonacci sequence (normalized to the $(m_i(X(n)), 1]$ interval).

If the selected action a_i results in good environment response ($0 \leq X(n) < m_i(X(n))$), then we reward this action, otherwise ($m_i(X(n)) < X(n) \leq 1$), we penalize it. The reward (penalty) parameters depend on how good (bad) the environment response was. Therefore, for each of the above environment responses, we use different reward rates $\alpha^r, r = 1, ..., R$ and penalty rates $\beta^p, p = 1, ..., P$, with $1 > \alpha^1 > \alpha^2 > ... > \alpha^R > 0$, and $1 > \beta^1 > \beta^2 > ... > \beta^P > 0$.

The above concepts produce the *Q-model Multiple Response (Q-MR) algorithm*:

Let $a(n) = a_i$

$$If\ X(n) = X_i^1(X(n)), then$$

$$P_i(n+1) = P_i(n) + g_i^1(X(n))[1 - P_i(n)]$$

$$P_j(n+1) = P_j(n) - g_i^1(X(n))P_j(n) \qquad \forall\ j \neq i$$

...

$$If\ X(n) = X_i^R(X(n)), then$$

$$P_i(n+1) = P_i(n) + g_i^R(X(n))[1 - P_i(n)]$$

$$P_j(n+1) = P_j(n) - g_i^R(X(n))P_j(n) \qquad \forall\ j \neq i$$

$$If\ X(n) = \bar{X}_i^P(X(n)), then$$

$$P_i(n+1) = P_i(n) - h_i^P(X(n))P_i(n)$$

$$P_j(n+1) = P_j(n) + h_i^P(X(n))\left[\frac{1}{|a|-1} - P_j(n)\right] \qquad \forall\ j \neq i$$

...

$$If\ X(n) = \bar{X}_i^1(X(n)), then$$

$$P_i(n+1) = P_i(n) - h_i^1(X(n))P_i(n)$$

$$P_j(n+1) = P_j(n) + h_i^1(X(n))\left[\frac{1}{|a|-1} - P_j(n)\right] \qquad \forall\ j \neq i$$

where $g_i^r(.)$, $h_i^p(.) \in (0,1)$ $r = 1, ..., R$ $p = 1, ..., P$ are nonnegative continuous functions.

Define $\mathbf{P}(n) = [P_1(n), ..., P_{|a|}(n)]^T$: vector of action probabilities.

Define $d_i^r = P[X(n) = X_i^r(n)/a(n) = a_i] \in (0,1)$ the probability for reward response r, when action a_i is selected, and $c_i^p = P[X(n) = \bar{X}_i^p(n)/a(n) = a_i] \in (0,1)$ the probability for penalty response p, when action a_i is selected, such that

$$\sum_{r=1}^{R} d_i^r + \sum_{p=1}^{P} c_i^p = 1$$

Define $M_0 = \frac{1}{|a|}\sum_{i=1}^{|a|}\left[\sum_{r=1}^{R} X_i^r d_i^r + \sum_{p=1}^{P} \bar{X}_i^p c_i^p\right]$.

The average penalty received by the automaton conditioned on $\mathbf{P}(n)$ is

$$\begin{aligned} M(n) &= E[X(n)/\mathbf{P}(n)] = \\ &= \sum_{i=1}^{|a|} E[X(n)/\mathbf{P}(n), a(n) = a_i]P_i(n) = \\ &= \sum_{i=1}^{|a|}\left[\sum_{r=1}^{R} X_i^r d_i^r + \sum_{p=1}^{P} \bar{X}_i^p c_i^p\right] P_i(n) \Rightarrow \end{aligned}$$

$$\lim_{n\to\infty} E[M(n)] = \sum_{i=1}^{|a|}\left[\sum_{r=1}^{R} X_i^r d_i^r + \sum_{p=1}^{P} \bar{X}_i^p c_i^p\right] \lim_{n\to\infty} E[P_i(n)]$$

Next, we prove that at each iteration of the *Q-MR algorithm*, the action probabilities are always non-negative and sum to 1.

Lemma : *feasibility*

The Q-model Multiple Response (Q-MR) algorithm preserves the feasibility of the action probability space.

Proof:

Let at time n the action probabilities are feasible and action a_i is selected, i.e.

$0 \leq P_i(n) \leq 1$, $\sum_{i=1}^{|a|} P_i(n) = 1$ and

let $a(n) = a_i$:

If $X(n) = X_i^r(X(n))$, then

$$\begin{aligned} P_i(n+1) = \;& g_i^r(X(n)) + P_i(n)[1 - g_i^r(X(n))] \geq 0 \\ & \textit{since } 0 < g_i^r(X(n)) < 1 \textit{ and } P_i(n) \geq 0, \end{aligned}$$

$$\begin{aligned} P_j(n+1) = \;& P_j(n)[1 - g_i^r(X(n))] \geq 0 \; \forall j \neq i \\ & \textit{since } g_i^r(X(n)) < 1 \textit{ and } P_j(n) \geq 0, \end{aligned}$$

$$\begin{aligned} P_i(n+1) = \;& g_i^r(X(n)) + P_i(n)[1 - g_i^r(X(n))] \leq g_i^r(X(n)) + [1 - g_i^r(X(n))] = 1, \\ & \textit{since } P_i(n) \geq 0, \end{aligned}$$

$$\begin{aligned} P_j(n+1) = \;& P_j(n)[1 - g_i^r(X(n))] \leq 1 - g_i^r(X(n)) < 1, \\ & \textit{since } P_j(n) \geq 0 \textit{ and } g * r_i(X(n)) > 0, \end{aligned}$$

$$\begin{aligned} \sum_{i=1}^{|a|} P_i(n+1) &= \sum_{i=1}^{|a|} P_i(n) + g_i^r(X(n))[1 - P_i(n)] - \sum_{j=1 j\neq i}^{|a|} g_i^r(X(n))P_j(n) \\ &= \sum_{i=1}^{|a|} P_i(n) = 1. \end{aligned}$$

If $X(n) = \bar{X}_i^p(X(n))$, then

$$
\begin{aligned}
P_i(n+1) = \quad & P_i(n)[1 - h_i^p(X(n))] \geq 0 \\
& since\ P_i(n) \geq 0\ and\ h_i^p(X(n)) < 1,
\end{aligned}
$$

$$
P_j(n+1) = \quad h_i^p(X(n))\frac{1}{|a|-1} + P_j(n)[1 - h_i^p(X(n))] \geq 0,
$$

$$
since\ |a| > 1,\ 0 < h_i^p(X(n)) < 1\ and\ P_j(n) \geq 0,
$$

$$
\begin{aligned}
P_i(n+1) = \quad & P_i(n)[1 - h_i^p(X(n))] \leq 1 - h_i^p(X(n)) < 1, \\
& since\ 0 \leq P_i(n) \leq 1\ and\ 0 < h_i^p(X(n)),
\end{aligned}
$$

$$
\begin{aligned}
P_j(n+1) = \quad & h_i^p(X(n))\frac{1}{|a|-1} + P_j(n)[1 - h_i^p(X(n))] \leq \\
& \leq h_i^p(X(n))\frac{1}{|a|-1} + [1 - h_i^p(X(n))] \leq \\
& \leq \frac{h_i^p(X(n))[2 - |a|] + |a| - 1}{|a|-1} \leq 1, \\
& since\ 0 \leq P_j(n) \leq 1,\ 0 < h_i^p(X(n)) < 1\ and\ |a| \geq 2,
\end{aligned}
$$

$$
\begin{aligned}
\sum_{i=1}^{|a|} P_i(n+1) &= \sum_{i=1}^{|a|} P_i(n) - h_i^p(X(n))P_i(n) + \sum_{j=1 j\neq i}^{|a|} h_i^p(X(n)) \left[\frac{1}{|a|-1} - P_j(n)\right] \\
&= \sum_{i=1}^{|a|} P_i(n) = 1
\end{aligned}
$$

□

Next, we prove that the *Q-MR algorithm* is not trapped in a specific action, i.e. no action is selected with probability 1. This is a desirable property for the problem that we consider, since the system conditions continuously change and even if an action is the best for a long time interval, it may not be always so. So, we like to give a chance to the other actions in case the have become better.

Lemma : <u>*non-absorbing*</u>

The Q-model Multiple Response (Q-MR) algorithm is non-absorbing.

Proof:

Let $a(n) = a_i$:

Since $\sum_{i=1}^{|a|} P_i(n) = 1$, not all $P_i(n)$'s are equal to 0. Therefore, $\exists\ j$ such that $P_j(n) \in (0, 1]$. Since the reward response r happens with nonzero reward probability $d_j^r \in (0, 1)$,

$P_j(n+1) = P_j(n) - g_j^r * P_j(n) < P_j(n)$ with positive probability $d_j^r > 0$.

Therefore,

$\mathbf{P}(n+1) \neq \mathbf{P}(n)$ with positive probability. □

For the special case of $f_i^r(.) = \theta * \alpha_i^r$ and $g_i^p(.) = \theta * \beta_i^p$, with $0 < \theta \leq 1,\ \ 0 < \alpha_i^r < 1,\ \ 0 < \beta_i^p < 1$, we have the *Q-model Multiple Response Linear (Q-MRL) algorithm:*

Let $a(n) = a_i$

$If\ X(n) = X_i^1(X(n)),\ then$

$$P_i(n+1) = P_i(n) + \theta\alpha_i^1[1 - P_i(n)]$$
$$P_j(n+1) = P_j(n) - \theta\alpha_i^1 P_j(n) \qquad \forall\ j \neq i$$

...

$If\ X(n) = X_i^R(X(n)),\ then$

$$P_i(n+1) = P_i(n) + \theta\alpha_i^R[1 - P_i(n)]$$
$$P_j(n+1) = P_j(n) - \theta\alpha_i^R P_j(n) \qquad \forall\ j \neq i$$

$If\ X(n) = \bar{X}_i^P(X(n)),\ then$

$$P_i(n+1) = P_i(n) - \theta\beta_i^P P_i(n)$$
$$P_j(n+1) = P_j(n) + \theta\beta_i^P \left[\frac{1}{|a|-1} - P_j(n)\right] \qquad \forall\ j \neq i$$

...

$If\ X(n) = \bar{X}_i^1(X(n)),\ then$

$$P_i(n+1) = P_i(n) - \theta\beta_i^1 P_i(n)$$
$$P_j(n+1) = P_j(n) + \theta\beta_i^1 \left[\frac{1}{|a|-1} - P_j(n)\right] \qquad \forall\ j \neq i$$

Next, we prove that at each step of the *Q-MRL algorithm*, we approach to the optimum action.

Theorem : stricly distance diminishing

The Q-model Multiple Response Linear (Q-MRL) algorithm with $\alpha_i^r = \alpha^r \ \forall i$ *and* $\beta_i^P = \beta \ \forall i$ *is strictly distance diminishing.*

Proof:

Let $\mathbf{P}(n)$ and $\mathbf{Q}(n)$ be two different trajectories of the action probabilities.

Let $a(n) = a_i$:

If $X(n) = X_i^r(X(n))$, then

$$P_i(n+1) = P_i(n) + \theta\alpha^r[1 - P_i(n)]$$
$$P_j(n+1) = P_j(n) - \theta\alpha^r P_j(n) \quad \forall j \neq i$$

$$Q_i(n+1) = Q_i(n) + \theta\alpha^r[1 - Q_i(n)]$$
$$Q_j(n+1) = Q_j(n) - \theta\alpha^r Q_j(n) \quad \forall j \neq i$$

Then

$$||\mathbf{P}(n+1) - \mathbf{Q}(n+1)|| = \left[\sum_{j=1}^{|a|} [P_j(n+1) - Q_j(n+1)]^2 \right]^{1/2} =$$

$$= \Big[[P_i(n) + \theta\alpha^r[1 - P_i(n)] - Q_i(n) - \theta\alpha^r[1 - Q_i(n)]]^2 +$$

$$+ \sum_{j=1 j\neq i}^{|a|} [P_j(n) - \theta\alpha^r P_j(n) - Q_j(n) + \theta\alpha^r Q_j(n)]^2 \Bigg]^{1/2} =$$

$$= \left[(1 - \theta\alpha^r)^2 \sum_{j=1}^{|a|} [P_j(n) - Q_j(n)]^2 \right]^{1/2} =$$

$$= (1 - \theta\alpha^r)||\mathbf{P}(n) - \mathbf{Q}(n)|| < ||\mathbf{P}(n) - \mathbf{Q}(n)||$$

since $0 < \theta < 1,\ 0 < \alpha^r < 1$.

If $X(n) = \bar{X}_i^p(X(n))$, then

$$P_i(n+1) = P_i(n) - \theta\beta^p P_i(n)]$$

$$P_j(n+1) = P_j(n) + \theta\beta^p \left[\frac{1}{|a|-1} - P_j(n) \right] \quad \forall j \neq i$$

$$Q_i(n+1) = Q_i(n) - \theta\beta^p Q_i(n)]$$

$$Q_j(n+1) = Q_j(n) + \theta\beta^p \left[\frac{1}{|a|-1} - Q_j(n) \right] \quad \forall j \neq i$$

Then

$$||\mathbf{P}(n+1) - \mathbf{Q}(n+1)|| = \left[\sum_{j=1}^{|a|} [P_j(n+1) - Q_j(n+1)]^2 \right]^{1/2} =$$

$$= \Big[P_i(n) - \theta\beta^p P_i(n) - Q_i(n) + \theta\beta^p Q_i(n)]^2 +$$

$$+ \sum_{j=1 j\neq i}^{|a|} \left[P_j(n) + \theta\beta^p \left[\frac{1}{|a|-1} - P_j(n) \right] - Q_j(n) - \theta\beta^p \left[\frac{1}{|a|-1} - Q_j(n) \right] \right]^2 \Bigg]^{1/2} =$$

$$= \left[(1-\theta\beta^p)^2 \sum_{j=1}^{|a|} [P_j(n) - Q_j(n)]^2 \right]^{1/2} =$$

$$=(1\text{-}\theta\beta^p)||\mathbf{P}(n) - \mathbf{Q}(n)|| < ||\mathbf{P}(n) - \mathbf{Q}(n)||$$

since $0 < \theta < 1,\ 0 < \alpha^r < 1$. □

Define also the Q-MR$L_{\alpha=\beta}$ *algorithm*, when $R = P$ and $\alpha_i^k = \beta_i^k \ \ k = 1, ..., R$, and the Q-MR$L_{\alpha=\epsilon\beta}$ *algorithm*, when $R = P$ and $\alpha_i^k = \epsilon\beta_i^k \ \ k = 1, ..., R$.

Next, we evaluate the conditional expectation of $P_i(n+1)$ given $P_i(n)$:

$$E[P_i(n+1)/P_i(n)] =$$

$$\sum_{r=1}^{R} [P_i(n) + \theta\alpha_i^r[1 - P_i(n)]]\, P_i(n)d_i^r +$$

$$+\sum_{p=1}^{P} [P_i(n) - \theta\beta_i^p P_i(n)]\, P_i(n)c_i^p +$$

$$+ \sum_{j=1\ j\neq i}^{|a|} \sum_{r=1}^{R} \left[P_i(n) - \theta\alpha_j^r P_i(n)\right] P_j(n)d_j^r +$$

$$+ \sum_{j=1\ j\neq i}^{|a|} \sum_{p=1}^{P} \left[P_i(n) + \theta\beta_j^p \left[\frac{1}{|a|-1} - P_i(n)\right]\right] P_j(n)c_j^p =$$

$$= P_i(n) + \theta P_i(n)\sum_{r=1}^{R}\left[\alpha_i^r[1 - P_i(n)]d_i^r - \sum_{j=1 j\neq i}^{|a|}\alpha_j^r P_j(n)d_j^r\right] +$$

$$+\theta\sum_{p=1}^{P}\left[\sum_{j=1 j\neq i}^{|a|}\beta_j^p\left[\frac{1}{|a|-1} - P_i(n)\right]P_j(n)c_j^p - \beta_i^p[P_i(n)]^2 c_i^p\right] =$$

$$= P_i(n) + \theta P_i(n)\sum_{r=1}^{R}\left[\alpha_i^r\sum_{j=1 j\neq i}^{|a|}P_j(n)d_i^r - \sum_{j=1 j\neq i}^{|a|}\alpha_j^r P_j(n)d_j^r\right] +$$

$$+\theta\sum_{p=1}^{P}\left[\sum_{j=1 j\neq i}^{|a|}\beta_j^p\left[\frac{1}{|a|-1} - P_i(n)\right]P_j(n)c_j^p - \beta_i^p[P_i(n)]^2 c_i^p\right] =$$

$$= P_i(n) + \theta P_i(n)\sum_{r=1}^{R}\sum_{j=1 j\neq i}^{|a|}P_j(n)(\alpha_i^r d_i^r - \alpha_j^r d_j^r) +$$

$$+\theta\sum_{p=1}^{P}\left[\sum_{j=1 j\neq i}^{|a|}\beta_j^p\left[\frac{1}{|a|-1} - P_i(n)\right]P_j(n)c_j^p - \beta_i^p[P_i(n)]^2 c_i^p\right]$$

For the *Q-MRL algorithm* with $\alpha_i^r = \alpha^r \quad \forall i \quad \forall r$ and $\beta_i^p = \beta^p \quad \forall i \quad \forall p$, we have

$$E[P_i(n+1)/P_i(n)] = P_i(n) + \theta P_i(n)\sum_{r=1}^{R}\alpha^r\sum_{j=1 j\neq i}^{|a|}P_j(n)(d_i^r - d_j^r) +$$

$$+\theta\sum_{p=1}^{P}\beta^p\sum_{j=1 j\neq i}^{|a|}\left[\left[\frac{1}{|a|-1} - P_i(n)\right]P_j(n)c_j^p - [P_i(n)]^2 c_i^p\right]$$

For the Q-MR$L_{\alpha=\beta}$ *algorithm* with $\alpha_i^r = \beta_i^r = \alpha^r \quad \forall i \quad \forall r$, we have

$$E[P_i(n+1)/P_i(n)] = P_i(n) +$$

$$\theta \sum_{k=1}^{R} \alpha^k \left[\sum_{j=1 j \neq i}^{|a|} \left[P_i(n) P_j(n) (d_i^k - d_j^k - c_j^k) + \frac{1}{|a|-1} P_j(n) c_j^k \right] - [P_i(n)]^2 c_i^k \right]$$

For the two-action ($|a| = 2$) *Q-MRL algorithm*, we have

$$\begin{aligned} E[P_1(n+1)/P_1(n)] &= P_1(n) + \theta P_1(n)[1 - P_1(n)] \sum_{r=1}^{R} (\alpha_1^r d_1^r - \alpha_2^r d_2^r) + \\ &+ \theta \sum_{p=1}^{P} \left[\beta_2^p [1 - P_1(n)]^2 c_2^p - \beta_1^p [P_1(n)]^2 c_1^p \right] \end{aligned}$$

For the two-action *Q-MRL algorithm* with $\alpha_i^r = \alpha^r \quad i = 1, 2 \quad \forall\, r$ and $\beta_i^p = \beta^p \quad i = 1, 2 \quad \forall\, p$, we have

$$\begin{aligned} E[P_1(n+1)/P_1(n)] &= P_1(n) + \theta P_1(n)[1 - P_1(n)] \sum_{r=1}^{R} \alpha^r (d_1^r - d_2^r) + \\ &+ \theta \sum_{p=1}^{P} \beta^p \left[[1 - P_1(n)]^2 c_2^p - [P_1(n)]^2 c_1^p \right] \end{aligned}$$

For the two-action Q-MR$L_{\alpha=\beta}$ *algorithm* with $\alpha_i^k = \beta_i^k = \alpha^k \quad i = 1, 2 \quad k = 1, ..., R$, we have

$$\begin{aligned} E[P_1(n+1)/P_1(n)] &= P_1(n) + \theta \sum_{k=1}^{R} \alpha^k \left[P_i(n)[1 - P_1(n)](d_1^k - d_2^k) + \right. \\ &+ \left. [1 - P_1(n)]^2 c_2^k - [P_1(n)]^2 c_1^k \right] \end{aligned}$$

If in addition $d_1^k + c_1^k = d_2^k + c_2^k \quad k = 1, ..., R$, then the above relation becomes

$$E[P_1(n+1)/P_1(n)] = P_1(n) +$$

$$+\theta \sum_{k=1}^{R} \alpha^k \Big[\; P_1(n)(d_1^k - d_2^k) - [P_1(n)]^2(d_1^k - d_2^k) -$$

$$-[P_1(n)]^2 c_1^k + c_2^k - 2P_1(n)c_2^k + [P_1(n)]^2 c_2^k \; \Big] =$$

$$= P_1(n) + \theta \sum_{k=1}^{R} \alpha^k \left[P_1(n)(d_1^k - d_2^k - 2c_2^k) + c_2^k \right] =$$

$$= P_1(n) - \theta \sum_{k=1}^{R} \alpha^k \left[P_1(n)(c_1^k + c_2^k) - c_2^k \right] \Rightarrow$$

$$\begin{aligned}
E[P_1(n+1)] &= E[P_1(n)] - \theta \sum_{k=1}^{R} \alpha^k \Big[\; E[P_1(n)](c_1^k + c_2^k) - c_2^k \; \Big] \\
&= [1 - \theta \sum_{k=1}^{R} \alpha^k (c_1^k + c_2^k)] E[P_1(n)] - \theta \sum_{k=1}^{R} \alpha^k c_2^k \Rightarrow \\
E[P_1(n)] &= \left[1 - \theta \sum_{k=1}^{R} \alpha^k (c_1^k + c_2^k) \right]^n P_1(0) - \\
&- \frac{1 - \left[1 - \theta \sum_{k=1}^{R} \alpha^k (c_1^k + c_2^k) \right]^n}{\theta \sum_{k=1}^{R} \alpha^k (c_1^k + c_2^k)} \theta \sum_{k=1}^{R} \alpha^k c_2^k
\end{aligned}$$

Finally, if

$$\left|1-\theta\sum_{k=1}^{R}\alpha^k(c_1^k+c_2^k)]\right|<1$$

then

$$\lim_{n\to\infty}E[P_1(n)]=\frac{\sum_{k=1}^{R}\alpha^k c_2^k}{\sum_{k=1}^{R}\alpha^k(c_1^k+c_2^k)}$$

Thus if $\sum_{k=1}^{R}\alpha^k c_2^k<\sum_{k=1}^{R}\alpha^k c_1^k$, i.e. the penalty probability for action a_2 is smaller than the action probability for action a_1, then $\lim_{n\to\infty}E[P_1(n)]<\lim_{n\to\infty}E[P_2(n)]$, i.e. on the average action a_2 is chosen asymptotically with a higher probability than action a_1. The following Theorems follow:

Theorem : *ergodic*

The Q-MR algorithm is ergodic and $\mathbf{P}(n)$ *converges in distribution to a random variable* $\mathbf{P}^*$ *independent of the initial probability* $\mathbf{P}(0)$.

Theorem : *expedient*

The Q-MRL$_{\alpha-\beta}$ algorithm with $A=B$, $\alpha^k=\beta^k$, $d_1^k+c_1^k=d_2^k+c_2^k$, *is expedient.*

Proof:

$$\lim_{n\to\infty}E[M(n)]=\sum_{i=1}^{|a|}\left[\sum_{r=1}^{R}X_i^r d_i^r+\sum_{p=1}^{P}\bar{X}_i^p c_i^p\right]\frac{\frac{1}{\sum_{k=1}^{R}\alpha^k c_i^k}}{\sum_{i=1}^{|a|}\frac{1}{\sum_{k=1}^{R}\alpha^k c_i^k}}<M_0$$

□

Rewritting the conditional expected difference of the probability P_i from step n to step $n+1$, we have

$$
\begin{aligned}
E[P_i(n+1) - P_i(n)/\mathbf{P}(n) = \mathbf{P}] \;=\; \theta \; & \left[\sum_{r-1}^{R} \alpha^r [1 - P_i] P_i d_i^r - \right. \\
& - \sum_{p=1}^{P} \beta^p [P_i]^2 c_i^p - \\
& - \sum_{r=1}^{R} \alpha^r P_i \sum_{j=1 j \neq i}^{|a|} P_j d_j^r + \\
& \left. + \sum_{p=1}^{P} \beta^p \left[\frac{1}{|a|-1} - P_i \right] \sum_{j=1 j \neq i}^{|a|} P_j c_j^p \right]
\end{aligned}
$$

Define the follwing functions:

$$
\begin{aligned}
W_i^R(\mathbf{P}) &= P_i \sum_{r=1}^{R} \alpha^r \sum_{j=1 j \neq i}^{|a|} P_j (d_i^r - d_j^r) \\
W_i^P(\mathbf{P}) &= \sum_{p=1}^{P} \beta^p \sum_{j=1 j \neq i}^{|a|} \left[\left[\frac{1}{|a|-1} - P_i \right] P_j c_j^p - [P_i]^2 c_i^p \right] \\
W_i(\mathbf{P}) &= W_i^R(\mathbf{P}) + W_i^P(\mathbf{P}), \quad \sum_{i=1}^{|a|} W_i(\mathbf{P}) = 0 \\
\mathbf{W}(\mathbf{P}) &= [W_1(\mathbf{P}), ..., W_{|a|}(\mathbf{P})]^T
\end{aligned}
$$

Then we can write the expectation of the conditional incremental action probabilities as:

$$E[\mathbf{P}(n+1) - \mathbf{P}(n)/\mathbf{P}(n) = \mathbf{P}] = \theta\mathbf{W}(\mathbf{P})$$

$$E[[\mathbf{P}(n+1) - \mathbf{P}(n)][\mathbf{P}(n+1) - \mathbf{P}(n)]^T/\mathbf{P}(n) = \mathbf{P}] = \theta^2\mathbf{W}'(\mathbf{P})$$

$$E[|\mathbf{P}(n+1) - \mathbf{P}(n)|^n/\mathbf{P}(n) = \mathbf{P}] = \theta^n\mathbf{W}''(\mathbf{P})$$

The above defined functions have also the following properties:

$\mathbf{W}(\mathbf{P})$ is twice continuously differentiable in S_r.
$\mathbf{W}'(\mathbf{P}) - \mathbf{W}(\mathbf{P}) * \mathbf{W}^T(\mathbf{P})$ is differentiable in S_r.

Next, we evaluate the probability of action a_1, when there are only two possible actions ($|a| = 2$):

$$P_1(n+1) = \begin{cases} P_1(n) + \theta\alpha^1[1 - P_1(n)] & if\ a(n) = a_1\ and\ reward\ response\ 1 \\ \ldots & \\ P_1(n) + \theta\alpha^R[1 - P_1(n)] & if\ a(n) = a_1\ and\ reward\ response\ R \\ & \\ P_1(n) - \theta\beta^P P_1(n) & if\ a(n) = a_1\ and\ penalty\ response\ P \\ \ldots & \\ P_1(n) - \theta\beta^1 P_1(n) & if\ a(n) = a_1\ and\ penalty\ response\ 1 \\ & \\ P_1(n) - \theta\alpha^1 P_1(n) & if\ a(n) = a_2\ and\ reward\ response\ 1 \\ \ldots & \\ P_1(n) - \theta\alpha^R P_1(n) & if\ a(n) = a_2\ and\ reward\ response\ R \\ & \\ P_1(n) + \theta\beta^P[1 - P_1(n)] & if\ a(n) = a_2\ and\ penalty\ response\ P \\ \ldots & \\ P_1(n) + \theta\beta^1[1 - P_1(n)] & if\ a(n) = a_2\ and\ penalty\ response\ 1 \end{cases}$$

Then the previously defined functions become:

$$W_1^R(P_1) = P_1 \sum_{r=1}^{R} \alpha^r (1 - P_1)(d_1^r - d_2^r)$$

$$W_1^P(P_1) = \sum_{p=1}^{P} \beta^p \left[(1 - P_1)^2 c_2^p - (P_1)^2 c_1^p\right]$$

The following Theorem characterizes the zeros of the function $W(\mathbf{P})$:

Theorem :

For the Q-MRL algorithm,

$\exists$ unique $Q_1, Q_2 \in (0,1)$ such that $W^P(Q_1) = 0$ and $W(Q_2) = 0$ and

i) $Q_2 > Q_1 > \frac{1}{2}$, when $\sum_{r=1}^{R} \alpha^r (d_1^r - d_2^r) > 0$ and $\sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) > 0$ or

ii) $Q_2 < Q_1 < \frac{1}{2}$, when $\sum_{r=1}^{R} \alpha^r (d_1^r - d_2^r) < 0$ and $\sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) < 0$

Proof:

$$W_1^P(0) = \sum_{p=1}^{P} \beta^p c_2^p > 0$$

$$W_1^P(\frac{1}{2}) = \frac{1}{4}\sum_{p=1}^{P} \beta^p (c_2^p - c_1^p)$$

$$W_1^P(1) = \sum_{p=1}^{P} \beta^p (-c_1^p) < 0$$

$$If \ \ \sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) = 0, \quad then \ \ W_1^P(Q_1) = 0 \ \ for \ \ Q_1 = \tfrac{1}{2}$$

$$if \sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) > 0, \ \ then \ \ W_1(Q_2) = 0 \ \ for \ \ Q_2 = Q_1 = \tfrac{1}{2}$$

$$If \ \ \sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) > 0, \quad then \ \ W_1^P(Q_1) = 0 \ \ for \ \ Q_1 > \tfrac{1}{2}$$

$$if \sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) > 0, \ \ then W_1(Q_2) = 0 \ \ for \ \ Q_2 = Q_1 = \tfrac{1}{2}$$

$$If \ \ \sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) = 0, \quad then \ \ W_1^P(Q_1) = 0 \ \ for \ \ Q_1 < \tfrac{1}{2}$$

$$if \sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) > 0, \ \ then \ \ W_1(Q_2) = 0 \ \ for \ \ Q_2 = Q_1 = \tfrac{1}{2}$$

Similarly for the other cases □

If in the algorithm, we replace β^p by $\epsilon\beta^p$, then $\mathbf{W}(\mathbf{P})$ and $\mathbf{W}^{P'}(\mathbf{P})$ get multiplied by ϵ and ϵ^2.

Define

$$\mathbf{W}(\epsilon,\mathbf{P}) = \mathbf{W}^R(\mathbf{P}) + \epsilon\mathbf{W}^P(\mathbf{P}) \quad 0 < \epsilon \leq 1$$

$$\mathbf{W}(1,\mathbf{P}) = \mathbf{W}(\mathbf{P})$$

Then the following Theorem follows:

Theorem :

For the Q-MRL algorithm $\exists\ Q(\epsilon) \in (0,1)$, *such that* $W(\epsilon, Q(\epsilon)) = 0$ *and*

i) $Q(\epsilon) \geq Q_2$ *and* $Q(\epsilon) \to 1$ *as* $\epsilon \to 0$,

when $\sum_{r=1}^{R} \alpha^r(d_1^r - d_2^r) > 0$ *and* $\sum_{p=1}^{P} \beta^p(c_2^p - c_1^p) > 0$ *or*

ii) $Q(\epsilon) \leq Q_2$ *and* $Q(\epsilon) \to 0$ *as* $\epsilon \to 0$,

when $\sum_{r=1}^{R} \alpha^r(d_1^r - d_2^r) < 0$ *and* $\sum_{p=1}^{P} \beta^p(c_2^p - c_1^p) < 0$

Proof:

i)

$$\mathbf{W}(\epsilon,\mathbf{P}) = \mathbf{W}(1,\mathbf{P}) - (1-\alpha)\mathbf{W}^P(\mathbf{P})$$

$$\mathbf{W}(1,\mathbf{P}) = \mathbf{W}(\mathbf{P})$$

Then

$$\mathbf{W}(\epsilon, Q_2) = -(1-\alpha)\mathbf{W}^P(Q_2) > 0$$

$$\mathbf{W}(\epsilon, 1) = 0$$

Therefore $\mathbf{W}(\epsilon, Q(\epsilon)) = 0 \quad Q(\epsilon) \geq Q_2$

If $\epsilon' > \epsilon$, then $\mathbf{W}(\epsilon', Q(\epsilon)) = \mathbf{W}^P(Q(\epsilon))(\epsilon' - \epsilon) > 0$

i.e. $Q(\epsilon)$ increases as ϵ decreases.

$\mathbf{W}(0,1) = 0$, thus 1 is the least upper bound on $Q(\epsilon)$.

Similarly for case ii). □

6.6.2 S-MR Learning Automata

In this section, we introduce an S-model MR learning automaton algorithm, for which the environment's response takes continuous values:

Let $a(n) = a_i$

reward response 1: $0 \leq X(n) \leq X_i^1(X(n))$

reward response 2: $X_i^1(X(n)) < X(n) \leq X_i^2(X(n))$

...

reward response R: $X_i^{R-1}(X(n)) < X(n) \leq m_i(X(n))$

penalty response P: $m_i(X(n)) < X(n) \leq \bar{X}_i^{P-1}(X(n))$

penalty response P-1: $\bar{X}_i^{P-1}(X(n)) < X(n) \leq \bar{X}_i^{P-2}(X(n))$

...

penalty response 1: $\bar{X}_i^1(X(n)) < X_i(n) \leq 1$

where $0 \leq X_i^1 < X_i^2 < ... < X_i^R < m_i << \bar{X}_i^{P-1} < ... < \bar{X}_i^1 \leq 1$ are functions of $X(n)$.

A possible sequence for these functions $\{X_i^r(X(n))\}$ could be a Fibonacci sequence (normalized to the $[0, m_i(X(n))$ interval). Also a possible sequence for the functions $\{\bar{X}_i^p(X(n))\}$ could be a Fibonacci sequence (normalized to the $(m_i(X(n)), 1]$ interval).

If the selected action a_i results in good environment response $(0 \leq X(n) < m_i(X(n)))$, then we reward this action, otherwise $(m_i(X(n)) < X(n) \leq 1)$, we penalize it. The reward (penalty) parameters depend on how good (bad) the environment response was. Therefore, for each of the above environment responses, we use different reward rates $\alpha^r, r = 1, ..., R$ and penalty rates $\beta^p, p = 1, ..., P$, with $1 > \alpha^1 > \alpha^2 > ... > \alpha^R > 0$, and $1 > \beta^1 > \beta^2 > ... > \beta^P > 0$.

The above concepts produce the *S-model Multiple Response (Q-MR) algorithm*:
Let $a(n) = a_i$

$If\ 0 \leq X(n) \leq X_i^1(X(n)), then$

$$P_i(n+1) = P_i(n) + g_i^1(X(n))[1 - P_i(n)]$$

$$P_j(n+1) = P_j(n) - g_i^1(X(n))P_j(n) \qquad \forall\ j \neq i$$

...

$If\ X_i^{R-1}(X(n)) < X(n) \leq m_i(X(n)), then$

$$P_i(n+1) = P_i(n) + g_i^R(X(n))[1 - P_i(n)]$$

$$P_j(n+1) = P_j(n) - g_i^R(X(n))P_j(n) \qquad \forall\ j \neq i$$

$If\ m_i(X(n)) < X(n) \leq \bar{X}_i^{P-1}(X(n)), then$

$$P_i(n+1) = P_i(n) - h_i^P(X(n))P_i(n)$$

$$P_j(n+1) = P_j(n) + h_i^P(X(n))\left[\frac{1}{|a|-1} - P_j(n)\right] \qquad \forall\ j \neq i$$

...

$If\ \bar{X}_i^1(X(n)) > X(n) \leq 1, then$

$$P_i(n+1) = P_i(n) - h_i^1(X(n))P_i(n)$$

$$P_j(n+1) = P_j(n) + h_i^1(X(n))\left[\frac{1}{|a|-1} - P_j(n)\right] \qquad \forall\ j \neq i$$

where $g_i^r(.)$, $h_i^p(.) \in (0,1)$ $\ r = 1, ..., R\ \ p = 1, ..., P$ are nonnegative continuous tunctions.

We can also prove several properties of the *S-MR* algorithm similar to those of the *Q-MR* algorithm.

Lemma : *feasibility*

The S-model Multiple Response (Q-MR) algorithm preserves the feasibility of the action probability space.

Lemma : *non-absorbing*

The S-model Multiple Response (Q-MR) algorithm is non-absorbing.

Theorem : strictly distance diminishing

The S-model Multiple Response Linear (Q-MRL) algorithm with $\alpha_i^r = \alpha^r \ \forall i$ *and* $\beta_i^p = \beta \ \forall i$ *is strictly distance diminishing.*

Theorem : *ergodic*

The S-MR algorithm is ergodic and $\mathbf{P}(n)$ *converges in distribution to a random variable* $\mathbf{P}^*$ *independent of the initial probability* $\mathbf{P}(0)$.

Theorem : *expedient*

The $S\text{-}MRL_{\alpha-\beta}$ *algorithm with* $A = B, \quad \alpha^k = \beta^k, \quad d_1^k + c_1^k = d_2^k + c_2^k,$ *is expedient.*

Theorem :

For the S-MRL algorithm $\exists\ Q(\epsilon) \in (0,1)$, *such that* $W(\epsilon, Q(\epsilon)) = 0$ *and*

i) $Q(\epsilon) \geq Q_2$ *and* $Q(\epsilon) \to 1$ *as* $\epsilon \to 0$,

when $\sum_{r=1}^{R} \alpha^r (d_1^r - d_2^r) > 0$ *and* $\sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) > 0$ *or*

ii) $Q(\epsilon) \leq Q_2$ *and* $Q(\epsilon) \to 0$ *as* $\epsilon \to 0$,

when $\sum_{r=1}^{R} \alpha^r (d_1^r - d_2^r) < 0$ *and* $\sum_{p=1}^{P} \beta^p (c_2^p - c_1^p) < 0$

6.7 Application to Datagram Networks

In this section, we propose using stochastic learning automata for load sharing, routing and congestion control decisions in datagram networks. Glorioso & Colon [195, 194], Srikantakumar [454, 453, 455] Chrystall, Mars & Narendra [104, 370] Nedzelnitsky & Narendra [349, 350], Mason [321, 322] and Narendra & Wheeler [345] use learning automata in datagram routing and they update the routing probabilities according to the delay experienced by a packet.

Learning automata have also been used for routing decisions in datagram networks with variable quality links [141]. When packets repeatedly fail transmission through a link, due to high error rate of this link, then they are driven by learning automata to use a different link. For a full description of learning automata-based routing in such an unreliable network, we refer to our paper [141].

The methodology that we propose for learning automata-based load sharing, routing and congestion control decisions for new arriving packets (in datagram networks) is similar to that of the next section for new arriving virtual circuits (in virtual circuit networks). So, we do not reiterate it here.

6.8 Application to Virtual Circuit Networks

In this section, we propose using stochastic learning automata for load sharing, routing and congestion control decisions in virtual circuit networks. We have introduced learning automata for virtual circuit routing [136], where the routing probabilities are updated according to the unfinished work on the selected path (for user optimum), or the increase in the number of packets on a path (or the increase in the portion of the overall network delay corresponding to this path) due to the addition of a new virtual circuit on this path (for system optimum). We considered three cases regarding the availability of traffic measurements: i) measurements of the number of virtual ciruits and packets on each path are known, ii) measurements of the number of virtual ciruits on each path are known, iii) no measurements of the network state are known, but the virtual circuit arrival rates are known.

Perturbation analysis may also be used in order to estimate the derivatives of the cost function.

Everything in this section is for each class c, but for easier exposition we do not show the superscript of the class. At every source node $[s.]$, a load sharing decision maker selects the destination node $[.d]$ where a new virtual circuit will be executed. Then a router selects the path $\pi_{[sd]}$ through which the virtual circuit will be transferred to the destination node $[.d]$ or rejects the virtual circuit for congestion control reasons. The length of a path is given by the Theorems of chapters 4 and 5.

However, network conditions change very rapidly and the minimum length path at a time instant may not be the same at the next time instant. Also, the information about the network state is always obsolete and inacurrate. Therefore the load sharing, routing and congestion control decisions should not overreact and immediately send a new virtual circuit to the estimated minimum length destination through the minimum length path, because oscillations may appear [45]. The system management decisions should fast track the current network state but without introducing instability.

The proposed adaptive routing algorithms are based on a "Probabilistic Selection of the Minimum Length Path" idea. Instead of using a definitive decision as to where to send a newly arriving virtual circuit, we vary the load sharing, routing and congestion control probabilities favoring the minimum length destination and path. Finally, the lengths of the paths are equalized.

Every source node $[s.]$ has a learning automaton for selecting a destination node where a new arriving virtual circuit will be executed. These learning automata operate asynchronously and base their decisions on the current system state. The actions, $a(n)$, of each automaton are the selection of a particular destination node $[.d]$ for processing the virtual circuit.

The automaton selects action $a(n) = a_{[sd]}$ with probability $P_{[sd]}(n)$ at each instant n. Action $a(n)$ becomes input to the environment. If this results in a favorable outcome for the network performance ($X(n) \rightarrow 0$), then the probability $P_{[sd]}(n)$ is increased by $\Delta P_{[sd]}(n)$ and the $P_{[sd']}(n)$, $\forall [sd'] \neq [sd]$, are decreased by

$\Delta P_{[sd']}(n)$. Otherwise, if an unfavorable outcome ($X(n) \rightarrow 1$) appears, then the $P_{[sd]}(n)$ is decreased by $\Delta P_{[sd]}(n)$ and the $P_{[sd']}(n)$, $\forall [sd'] \neq [sd]$ are increased by $\Delta P_{[sd']}(n)$.

We propose the following adaptive algorithm at every source node $[s.]$, for the load sharing decisions:

Probabilistic Selection of the Minimum Length Destination:

Suppose destination $[.d]$ was selected at time n-1, with $P_{[sd]}(n-1)$.

Compute the lengths to all destinations, $l_{[sd']}(n)$ $\forall [sd']$

Calculate $X(n) = X(\ldots, l_{[sd']}(n), \ldots)$.

Update the load sharing probabilities $P_{[sd']}(n)$ $\forall [sd']$.

Select the destination for the n^{th} virtual circuit probabilistically according to $P_{[sd']}(n)$.

Similarly, for the routing and congestion control decisions, every source node $[s.]$ has a learning automaton for every destination node $[.d]$ that routes new arriving virtual circuits at node $[s.]$ and destined for node $[.d]$. These learning automata operate asynchronously and base their decisions on the current network state. The actions, $a(n)$, of each automaton are to select some particular path $\pi[sd]$ to the destination node $[.d]$.

The automaton selects action $a(n) = a_{\pi[sd]}$ with probability $P_{\pi[sd]}(n)$ at each instant n. Action $a(n)$ becomes input to the environment. If this results in a favorable outcome for the network performance ($X(n) \rightarrow 0$), then the probability $P_{\pi[sd]}(n)$ is increased by $\Delta P_{\pi[sd]}(n)$ and the $P_{p[sd]}(n)$, $\forall p[sd] \neq \pi[sd]$, are decreased by $\Delta P_{p[sd]}(n)$. Otherwise, if an unfavorable outcome ($X(n) \rightarrow 1$) appears, then the $P_{\pi[sd]}(n)$ is decreased by $\Delta P_{\pi[sd]}(n)$ and the $P_{p[sd]}(n)$, $\forall p[sd] \neq \pi[sd]$ are increased by $\Delta P_{p[sd]}(n)$.

We propose the following adaptive algorithm at every source node $[s.]$, for routing virtual circuits to a certain destination node $[.d]$ or for rejecting them when congestion exists into the network:

Probabilistic Selection of the Minimum Length Path:

Suppose path $\pi[sd]$ *was selected at time n-1, with* $P_{\pi[sd]}(n-1)$.

Compute all paths lengths, $l_{p[sd]}(n)\ \forall p[sd] and l_{o[sd]}(n)$.

Compute $X(n) = X(..., l_{p[sd]}(n), ...)$.

Update the routing probabilities $P_{p[sd]}(n)\ \forall p[sd]$, $P_{o[sd]}(n)$.

Select the path for the n^{th} *virtual circuit or reject it probabilistically according to* $P_{p[sd]}(n)$, $P_{o[sd]}(n)$.

6.8.1 Simulation Comparison of Algorithms

In this section, we apply three learning automata algorithms to the routing problem in virtual circuits networks and we compare their performance. All algorithms have the same reward and penalty parameters for different actions and the parameter $\theta = 1$. We consider as length of a path $\pi[sd]$ its average packet delay $T_{\pi[sd]}(n)$.

The simplest information that someone can measure and transfer about the network state is the packet delay through each path from source to destination. A new arriving virtual circuit is routed from its source to its destination along the path that promises the minimum packet delay. Instead of using a definitive decision as to where to send a newly arriving virtual circuit, we vary the path routing probabilities favoring the minimum delay path.

The first algorithm is the $L_{R-\epsilon P}$ learning automaton with reward parameter $\alpha = 0.2$ and penalty parameter $\beta = 0.8$. If the selected path has the minimum packet delay at the next iteration, then we increase the probability of selecting it again, otherwise we decrease it. More specifically:

Let path $\pi[sd]$ is selected at time n

$$
\begin{aligned}
&If\ T_{\pi[sd]}(n) = \min_{p[sd] \in \Pi_{[sd]}} \{T_{p[sd]}(n)\},\ then\\
&\quad P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) + 0.2 * [1 - P_{\pi[sd]}(n)]\\
&\quad P_{p[sd]}(n+1) = P_{p[sd]}(n) - \ 0.2 * P_{p[sd]}(n) \qquad \forall\ p[sd] \neq \pi[sd]\\
&else\\
&\quad P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) - 0.8 * P_{\pi[sd]}(n)\\
&\quad P_{p[sd]}(n+1) = P_{p[sd]}(n) + 0.8 * \left[1 - P_{p[sd]}(n)\right] \qquad \forall\ p[sd] \neq \pi[sd]
\end{aligned}
$$

The second algorithm is the $MRL_{R-\epsilon P}$ learning automaton with reward parameters $\alpha^1 = 0.8$ (excellent choice), $\alpha^2 = 0.2$ (good choice, but not excellent) and penalty parameters $\beta^2 = 0.8$ (bad choice), $\beta^1 = 1$ (very bad choice). We consider two response and penalty regions for the algorithm ($P = R = 2$) and the functions that define these regions are linear functions with parameter 2. More specifically:

Let path $\pi[sd]$ is selected at time n

$$If\ T_{\pi[sd]}(n) \leq \min_{p[sd]\in\Pi_{[sd]}} \{T_{p[sd]}(n)/2\},\ then$$
$$P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) + 0.8 * [1 - P_{\pi[sd]}(n)]$$
$$P_{p[sd]}(n+1) = P_{p[sd]}(n) - 0.8 * P_{p[sd]}(n) \qquad \forall\ p[sd] \neq \pi[sd]$$

$$If \min_{p[sd]\in\Pi_{[sd]}} \{T_{p[sd]}(n)/2\} < T_{\pi[sd]}(n) \leq \min_{p[sd]\in\Pi_{[sd]}} \{T_{p[sd]}(n)\},$$
$$P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) + 0.2 * [1 - P_{\pi[sd]}(n)]$$
$$P_{p[sd]}(n+1) = P_{p[sd]}(n) - 0.2 * P_{p[sd]}(n) \qquad \forall\ p[sd] \neq \pi[sd]$$

$$If \min_{p[sd]\in\Pi_{[sd]}} \{T_{p[sd]}(n)\} \leq T_{\pi[sd]}(n) \leq \min_{p[sd]\in\Pi_{[sd]}} \{2 * T_{p[sd]}(n)\},$$
$$P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) - 0.8 * P_{\pi[sd]}(n)$$
$$P_{p[sd]}(n+1) = P_{p[sd]}(n) + 0.8 * \left[1 - P_{p[sd]}(n)\right] \qquad \forall\ p[sd] \neq \pi[sd]$$

$$If \min_{p[sd]\in\Pi_{[sd]}} \{2 * T_{p[sd]}(n)\} \leq T_{\pi[sd]}(n),$$
$$P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) - 1 * P_{\pi[sd]}(n)$$
$$P_{p[sd]}(n+1) = P_{p[sd]}(n) + 1 * \left[1 - P_{p[sd]}(n)\right] \qquad \forall\ p[sd] \neq \pi[sd]$$

Finally, the third algorithm is the $SDL_{R-\epsilon P}$ learning automaton with reward parameter $\alpha = 0.2$ and penalty parameter $\beta = 0.8$. The state dependent parameter is an exponential function of the difference of the selected path average delay and the maximum average delay of paths between this source-destination. More specifically:

Let path $\pi[sd]$ is selected at time n

$$If\ T_{\pi[sd]}(n) = \min_{p[sd]\in\Pi_{[sd]}} \{T_{p[sd]}(n)\},\ then$$

$$P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) + 0.2 * (1 - e^{T_{\pi[sd]}(n) - \max_{p[sd]} T_{p[sd]}}) * [1 - P_{\pi[sd]}(n)]$$

$$P_{p[sd]}(n+1) = P_{p[sd]}(n) - 0.2 * (1 - e^{T_{\pi[sd]}(n) - \max_{p[sd]} T_{p[sd]}}) * P_{p[sd]}(n)$$

$$\forall\ p[sd] \neq \pi[sd]$$

$else$

$$P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) - 0.8 * (1 - e^{T_{\pi[sd]}(n) - \max_{p[sd]} T_{p[sd]}}) * P_{\pi[sd]}(n)$$

$$P_{p[sd]}(n+1) = P_{p[sd]}(n) + 0.8 * (1 - e^{T_{\pi[sd]}(n) - \max_{p[sd]} T_{p[sd]}}) * \left[1 - P_{p[sd]}(n)\right]$$

$$\forall\ p[sd] \neq \pi[sd]$$

It is important to understand that the above values for the reward and penalty parameters are not the optimum. Depending on the network topology, the number of paths between source-destination pairs, the traffic characteristics, the information about the network state, the updating time interval and other variables, we should choose the best parameters by experimentation. Note that the traditionally used shortest path algorithm is a special case of the learning automata algorithm, since by suitable tuning the parameters, we can select the minimum length path with probability 1.

In this section, we compare the performance of the three learning automata algorithms (see previous section) via simulation. We consider a network with two paths from source to destination. So, each learning automaton has two actions $\|a\| = 2$ to choose. Path # 1 has seven links with service rates 1. Path # 2 has seven links with service rates 1, 0.5, 2, 2, 2, 0.5 and 1 (Figure 6.2).

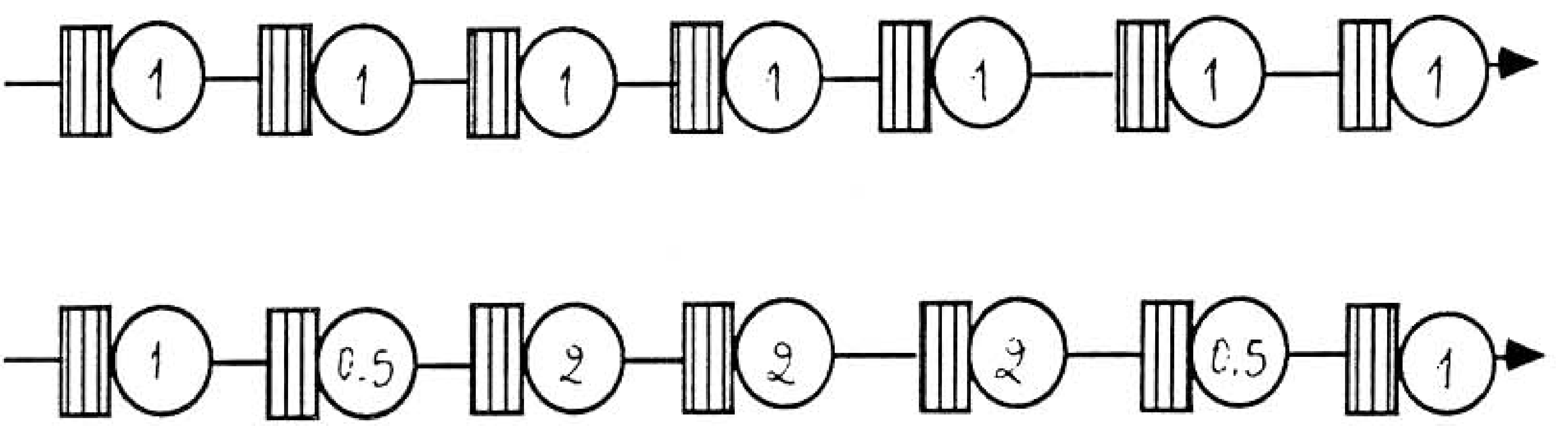

Figure 6.2: Simulated network.

The mean packet service requirement is $1/\mu = 1$ and therefore $\mu_{ij} = \mu * C_{ij} = C_{ij}$. For the rest traffic characteristics, we considered two cases:

i) 30/2/40: the virtual circuit arrival rate is $\gamma = 1/30$, the packet arrival rate per virtual circuit is $r = 1/2$ and the mean virtual circuit duration is $\delta = 40$.

ii) 50/5/200: the virtual circuit arrival rate is $\gamma = 1/50$, the packet arrival rate per virtual circuit is $r = 1/5$ and the mean virtual circuit duration is $\delta = 200$.

For measuring the path delay, we consider two cases:

i) 1 : at every packet departure from the network through a path, the destination sends to the source the packet delay through the path of that last packet.

ii) 50 : at every 50 packet departure from the network through a path, the destination sends to the source the average packet delay through the path of these 50 last packets.

The source node keeps and updates the information about the delay of its paths to the destination. The information about the delay of a path is updated every time a packet arrives at the destination through this path. However, this updating is not done immediately, but we assume a feedback delay so that this information becomes available to the source node. We assume that no extra traffic is created for transferring this feedback information to the source node (it is either piggybacked on regular packets or uses a different channel). We consider two cases:

i) *instantaneous* information, when the feedback delay is 7 time units. In this case, we assume that the feedback information has higher priority over other packets and does not wait in queues.

ii) *obsolete* information, when the feedback delay is 60 time units. In this case, we assume that the feedback information is piggybacked on regular packets and is transferred back to the source node.

Updating the information of a path asynchronously at packet departure instances has an undesirable characteristic. If a path becomes unattractive for routing packets through it, then we may not route any more packets through it. However, our information about its length remains the same, although after some time this path may become idle. We have overcome this problem by sending a

30/2/40	1 instant	1 obsolete	50 instant	50 obsolete
deterministic	50.59 ±0.89	63.59 ±1.28	55.38 ±0.93	61.97 ±1.36
L automaton	50.27 ±1.15	61.29 ±1.36	57.37 ±0.88	61.44 ±1.25
MRL automaton	50.64 ±0.73	61.27 ±1.63	61.15 ±0.92	64.04 ±1.36
SDL automaton	48.92 ±0.51	62.52 ±1.04	57.37 ±1.14	60.60 ±1.46

50/5/200	1 instant	1 obsolete	50 instant	50 obsolete
deterministic	46.79 ±1.75	57.84 ±1.92	60.52 ±2.21	68.30 ±2.23
L automaton	45.35 ±1.45	54.85 ±2.31	61.43 ±1.76	65.43 ±1.77
MRL automaton	43.25 ±1.45	56.45 ±2.13	62.05 ±3.16	65.67 ±2.52
SDL automaton	46.22 ±1.36	57.45 ±2.17	60.81 ±1.79	67.24 ±1.68

Table 6.1: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 for deterministic, Linear automaton, Multiple Response automaton and State Dependent automaton based routing.

probe packet through a path that has not been used for 100 time units and therefore updating our information about its delay.

In Figures 6.3-6.10 and Table 6.1, we show the simulation results for the average packet delay for 10,000 virtual circuits.

Although the reward and penalty parameters of the learning automata were not chosen to be the best possible, all four algorithms achieve similar performance. However, the learning automata have more flexibility, since we can calibrate their parameters depending on the particular system. Note, that the deterministic algorithm is a special case of the learning automata, since we can choose their parameters, such that the minimum length path is chosen. The more frequent we update the algorithms and the more recent state information we have, the better the performance.

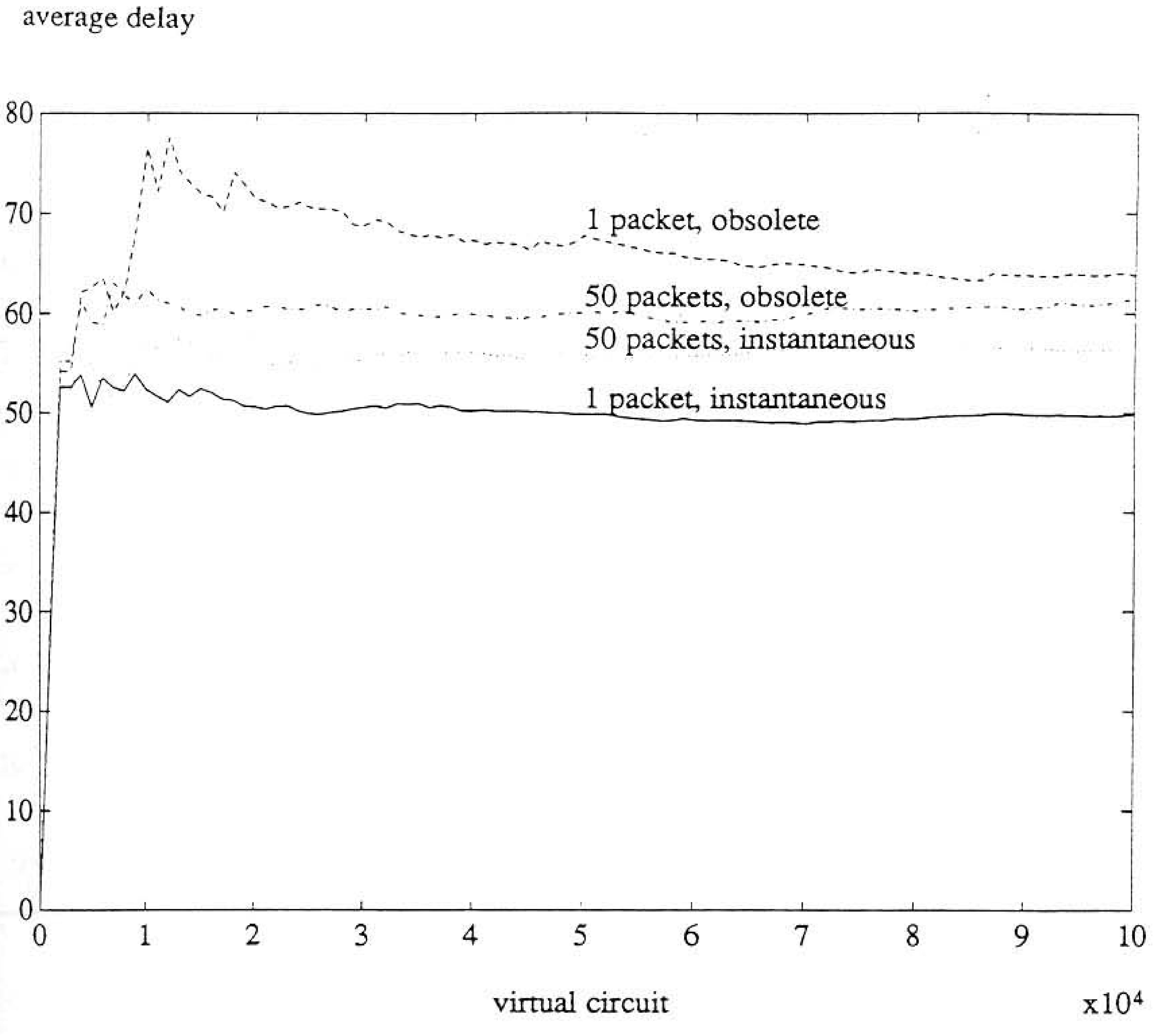

Figure 6.3: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/30$ for deterministic routing.

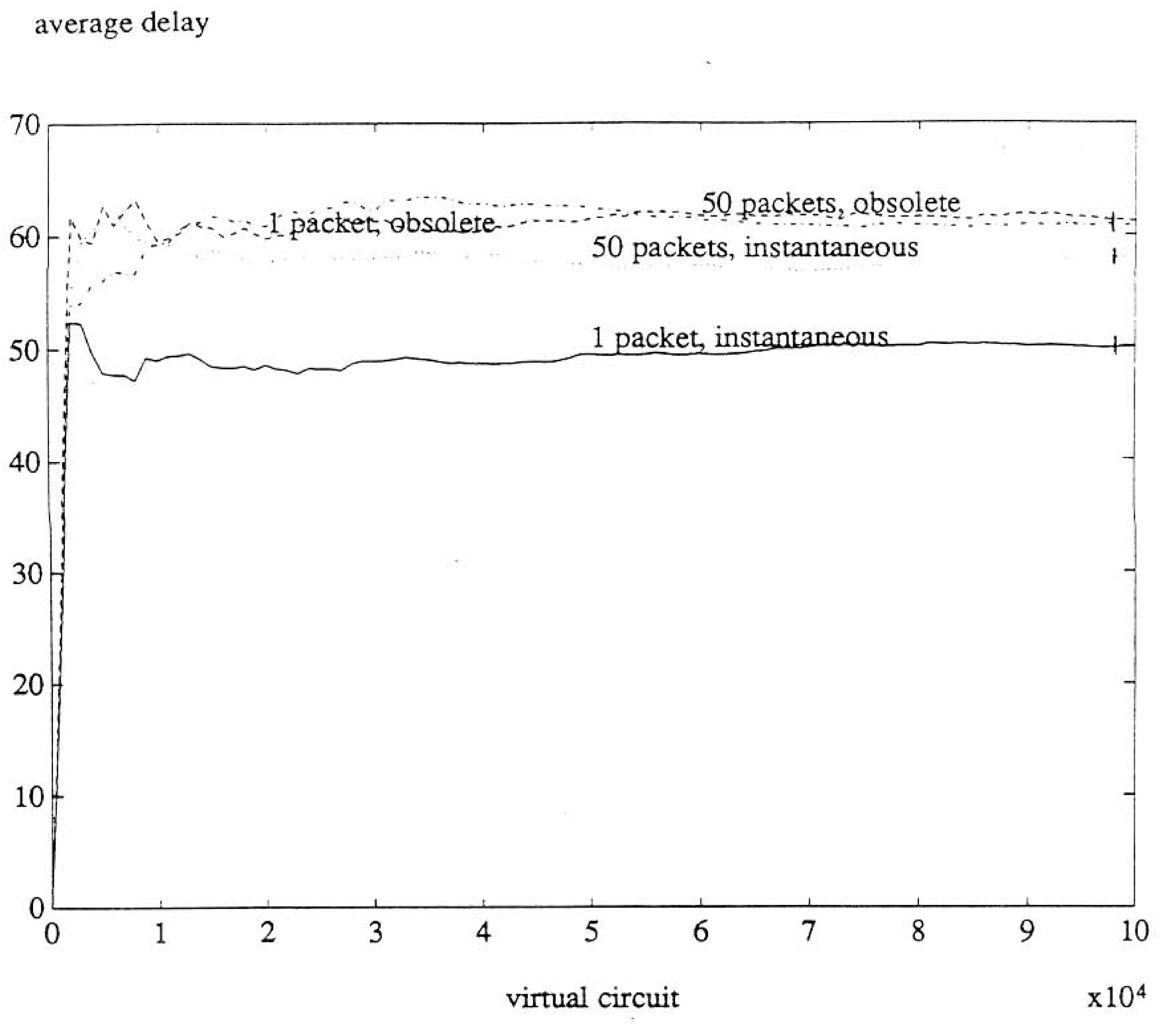

Figure 6.4: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/30$ for Linear learning automaton based routing.

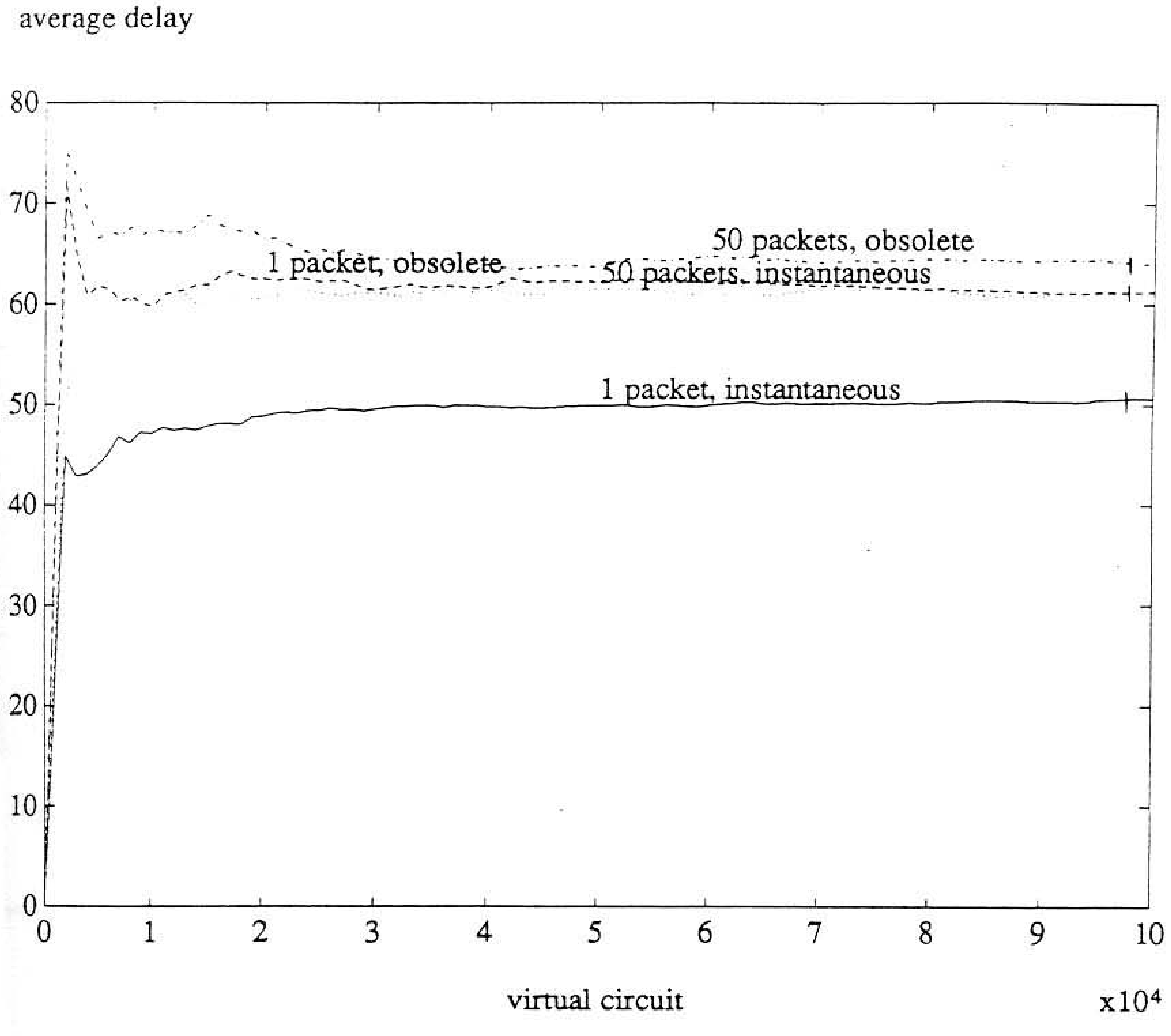

Figure 6.5: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/30$ for Multiple Response learning automaton based routing.

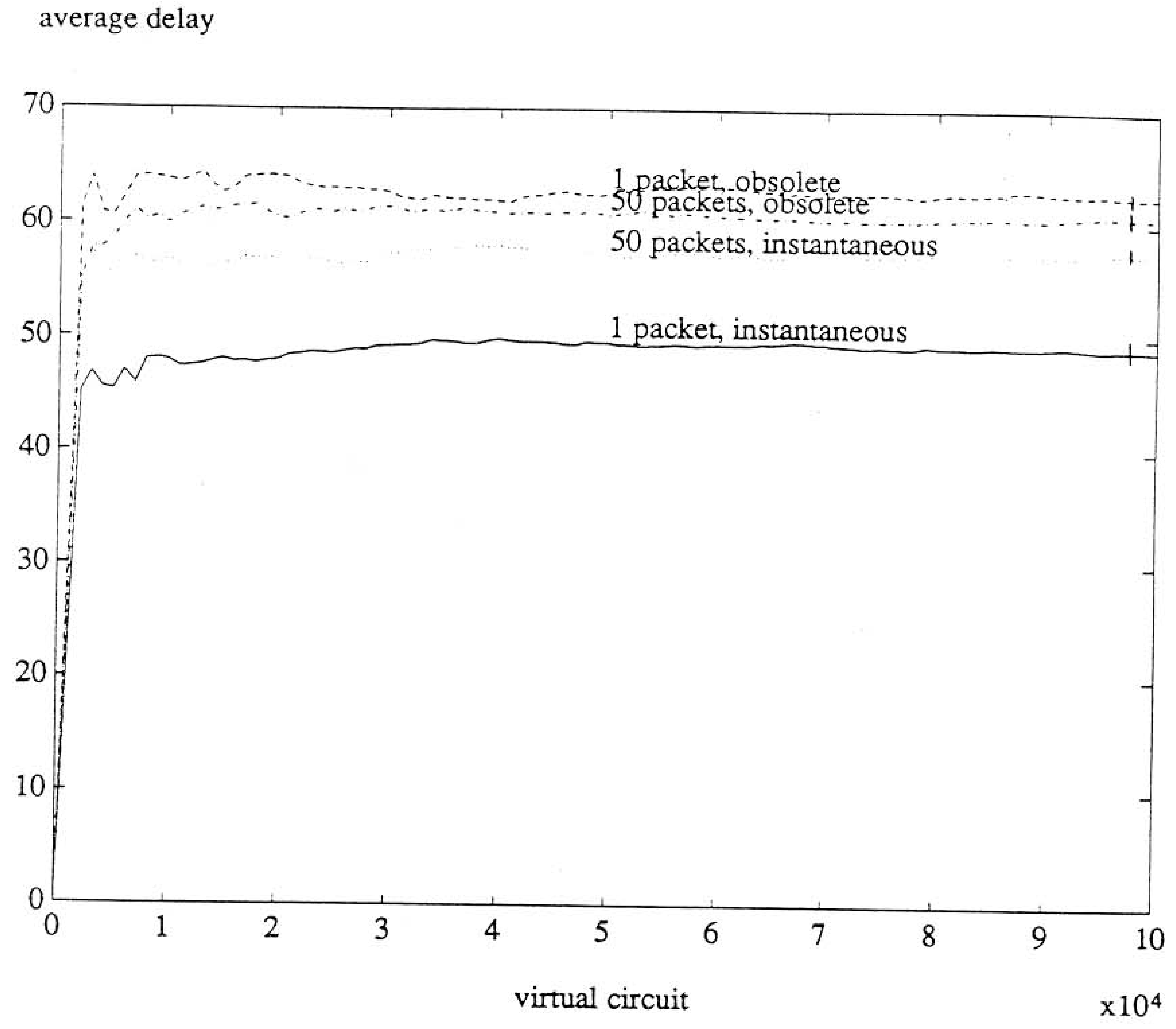

Figure 6.6: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/30$ for State Dependent learning automaton based routing.

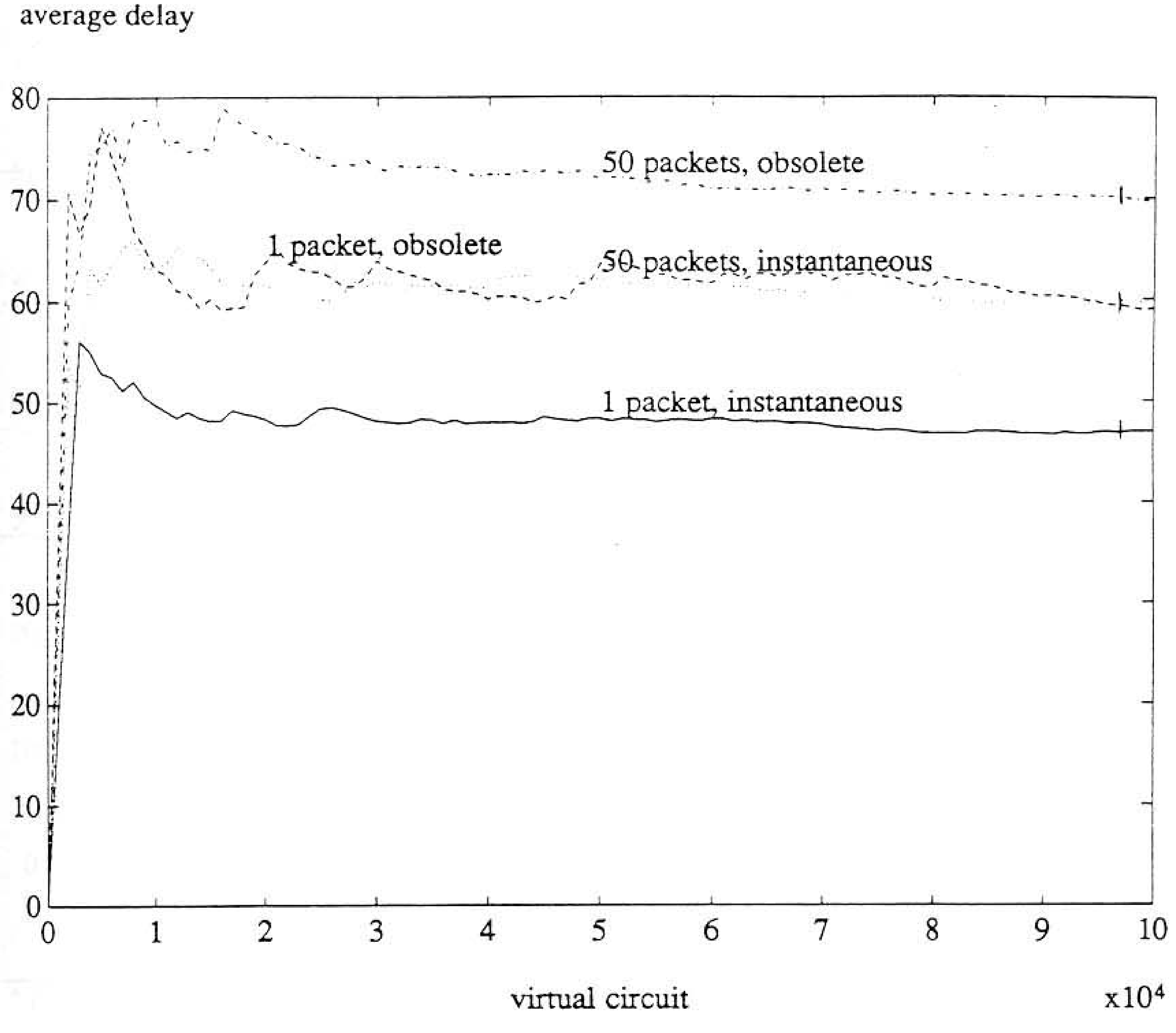

Figure 6.7: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/50$ for deterministic routing.

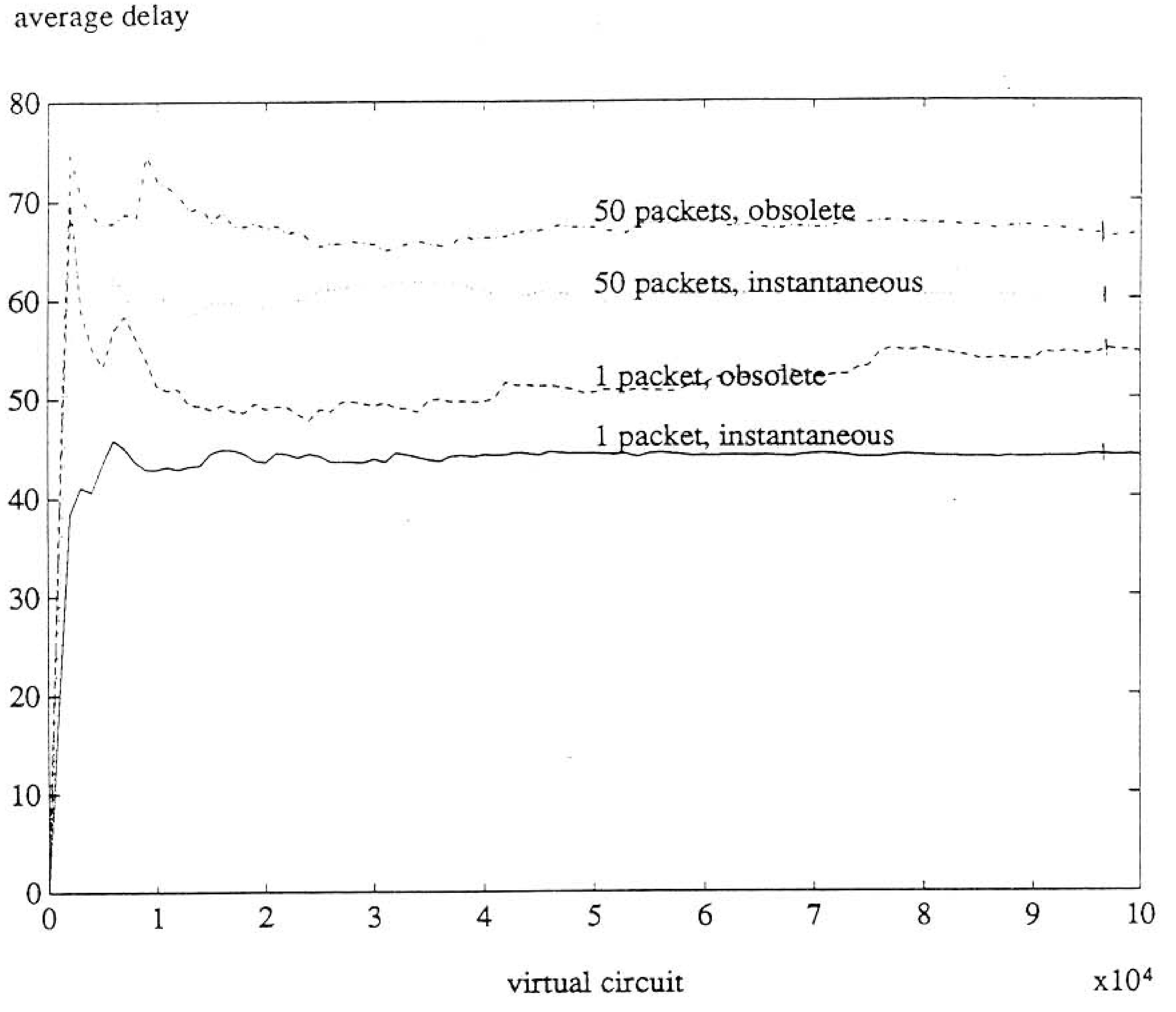

Figure 6.8: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/50$ for Linear learning automaton based routing.

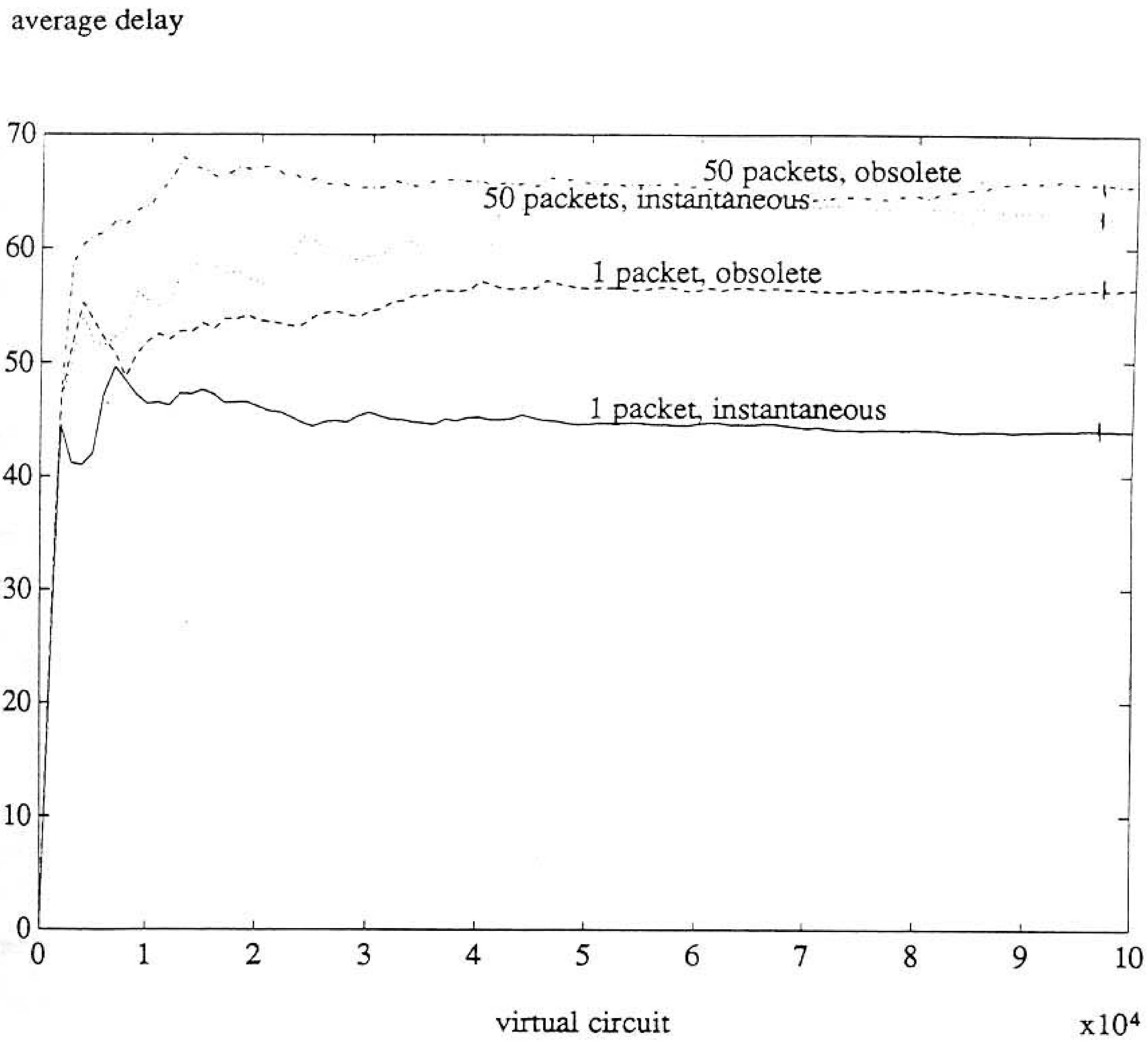

Figure 6.9: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/50$ for Multiple Response learning automaton based routing.

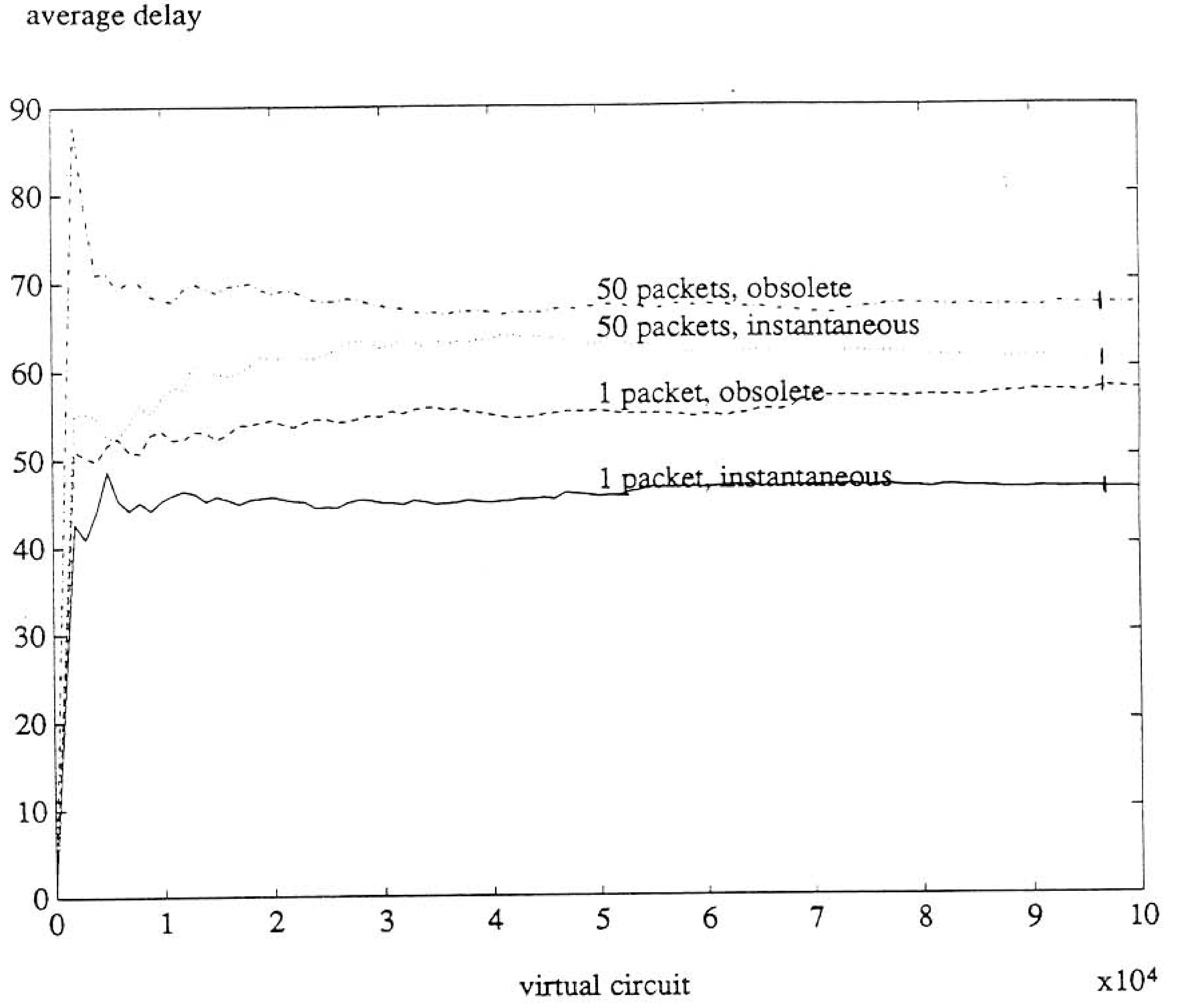

Figure 6.10: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 with $\gamma = 1/50$ for State Dependent learning automaton based routing.

6.8.2 Simulation Comparison of Performance Measures

In this section, we use the $L_{R-\epsilon P}$ algorithm. We compare a class of performance measures, that we introduced in section 5.5.3.

We consider as link length $l_{ij}(n)$ a convex combination of its current length $l_{ij}^{current}(n)$ and its future length $l_{ij}^{future}(n)$. We consider as current length $l_{ij}^{current}(n) = \dfrac{1+N_{ij}(n)}{\mu C_{ij}}$, a linear function of the number of packets on link ij. We consider as future length $l_{ij}^{current}(n) = \dfrac{1+V_{ij}(n)}{\mu C_{ij}}$, a linear function of the number of virtual circuits on link ij. Then the length of link ij is

$$l_{ij} = \epsilon * \frac{1+N_{ij}(n)}{\mu C_{ij}} + (1-\epsilon) * \frac{1+V_{ij}(n)}{\mu C_{ij}} \quad 0 \leq \epsilon \leq 1$$

A special case of this measure is the unfinished work [136]

$$U_{ij}(n) = \frac{1+N_{ij}(n)}{\mu C_{ij}} + \frac{r}{\delta} * \frac{1+V_{ij}(n)}{\mu C_{ij}}$$

The length of a path $\pi[sd]$ is

$$l_{\pi[sd]}(n) = \sum_{ij} l_{ij}(n) * 1_{ij \in \pi[sd]}(n)$$

Next, we investigate the effect of the parameter ϵ on the average packet delay.

The routing decisions are done by a $L_{R-\epsilon P}$ algorithm with reward parameter $\alpha = 0.2$ and penalty parameter $\beta = 0.8$: If the selected path has the minimum packet delay at the next iteration, then we increase the probability of selecting it again, otherwise we decrease it.

Let path $\pi[sd]$ is selected at time n

$$
\begin{aligned}
&If\ l_{\pi[sd]}(n) = \min_{p[sd] \in \Pi_{[sd]}} \{l_{p[sd]}(n)\},\ then \\
&\quad P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) + 0.2 * [1 - P_{\pi[sd]}(n)] \\
&\quad P_{p[sd]}(n+1) = P_{p[sd]}(n) - 0.2 * P_{p[sd]}(n) \qquad \forall\ p[sd] \neq \pi[sd] \\
&else \\
&\quad P_{\pi[sd]}(n+1) = P_{\pi[sd]}(n) - 0.8 * P_{\pi[sd]}(n) \\
&\quad P_{p[sd]}(n+1) = P_{p[sd]}(n) + 0.8 * \left[1 - P_{p[sd]}(n)\right] \qquad \forall\ p[sd] \neq \pi[sd]
\end{aligned}
$$

We consider the same network as that of the previous section The mean packet service requirement is $1/\mu = 1$ and therefore $\mu_{ij} = \mu * C_{ij} = C_{ij}$. The total packet arrival rate is $r * \gamma/\delta = 4/5$. Two cases that achieve this rate are the following:

i) 5/50/200: the virtual circuit arrival rate is $\gamma = 1/5$, the packet arrival rate per virtual circuit is $r = 1/50$ and the mean virtual circuit duration is $\delta = 200$.

ii) 50/5/200: the virtual circuit arrival rate is $\gamma = 1/50$, the packet arrival rate per virtual circuit is $r = 1/5$ and the mean virtual circuit duration is $\delta = 200$.

For measuring the path length, we consider two cases:

i) 1 : the current number of packets at each link is sent to the source at every packet departure from that link.

ii) 50 : the average number of packets at each link during the last 50 time units is sent to the source at every 50th packet departure from that link.

The source node keeps and updates the information about the delay of its paths to the destination. The information about the delay of a path is updated every time a packet arrives at the destination through this path. However, this updating is not done immediately, but we assume a feedback delay so that this information becomes available to the source node. We assume that no extra traffic is created for th transferring this feedback information to the source node (it is either piggybacked on regular packets or uses a different channel). We consider two cases:

i) *instantaneous* information, when the feedback delay is 7 time units. In this case, we assume that the feedback information has higher priority over other packets and does not wait in queues.

ii) *obsolete* information, when the feedback delay is 60 time units. In this case, we assume that the feedback information is piggybacked on regular packets and is transferred back to the source node.

In Figure 6.11, 6.12 and Table 6.2, we show the simulation results for the average packet delay for 10,000 virtual circuits.

We notice that a proper value for the parameter ϵ should be experimentally selected for best performance. Using only the number of virtual circuits on each link ($\epsilon = 0$) as the link length is very inefficient (actually, for the case 5/50/200,

5/50/200	1 instant	1 obsolete	50 instant	50 obsolete
$\epsilon = 0.2$	104.22 ±4.50	102.20 ±5.51	133.30 ±5.98	129.71 ±4.92
$\epsilon = 0.4$	59.61 ±3.31	59.97 ±3.06	78.49 ±2.75	73.94 ±2.38
$\epsilon = 0.6$	46.98 ±2.43	46.12 ±1.79	60.88 ±1.67	56.81 ±1.53
$\epsilon = 0.8$	39.77 ±1.05	42.68 ±1.25	64.12 ±2.05	77.94 ±3.33
$\epsilon = 1$	37.19 ±1.22	50.66 ±2.06	104.38 ±4.36	126.66 ±4.45
delay	55.97 ±3.98	97.02 ±8.79	106.41 ±8.03	121.67 ±8.15

50/5/200	1 instant	1 obsolete	50 instant	50 obsolete
$\epsilon = 0$	73.01 ±3.89	69.24 ±5.15	125.73 ±13.88	100.86 ±7.56
$\epsilon = 0.2$	36.70 ±0.98	37.20 ±0.83	51.43 ±1.66	51.50 ±1.77
$\epsilon = 0.4$	34.39 ±1.05	37.23 ±1.42	64.44 ±1.69	68.90 ±1.71
$\epsilon = 0.6$	34.85 ±1.05	39.41 ±1.10	76.96 ±1.50	85.60 ±1.10
$\epsilon = 0.8$	35.29 ±0.88	41.59 ±1.11	83.39 ±1.19	92.28 ±2.13
$\epsilon = 1$	37.02 ±1.02	44.15 ±0.99	86.26 ±2.34	97.26 ±3.46
delay	45.35 ±1.45	54.85 ±2.31	61.43 ±1.76	65.43 ±1.77

Table 6.2: The average packet delay ± error (95% confidence interval) for the network of Figure 6.2 for different values of the parameter ϵ, when we use as link length $l_{ij} = \epsilon * \frac{1 + N_{ij}}{ij} + (1 - \epsilon) * \frac{1 + V_{ij}}{ij}$.

the average network delay becomes extremely high and we do not even show it). Also, it is not always best to use only the number of packets on each link ($\epsilon = 1$) as the link length.

For comparison, we also show the average network delay, when we use the path delay as path length. It seems that using both the number of packets and virtual circuits as path length is much better than using the path delay.

The more frequent we update the algorithms and the more recent state information we have, the better the performance. Note also, that although the traffic characteristics 5/50/200 and 50/5/200 have the same packet arrival rate, the overall average packet delay is different. Traffic 5/50/200 has higher average packet delay than traffic 50/5/200, because the virtual circuits are arriving more

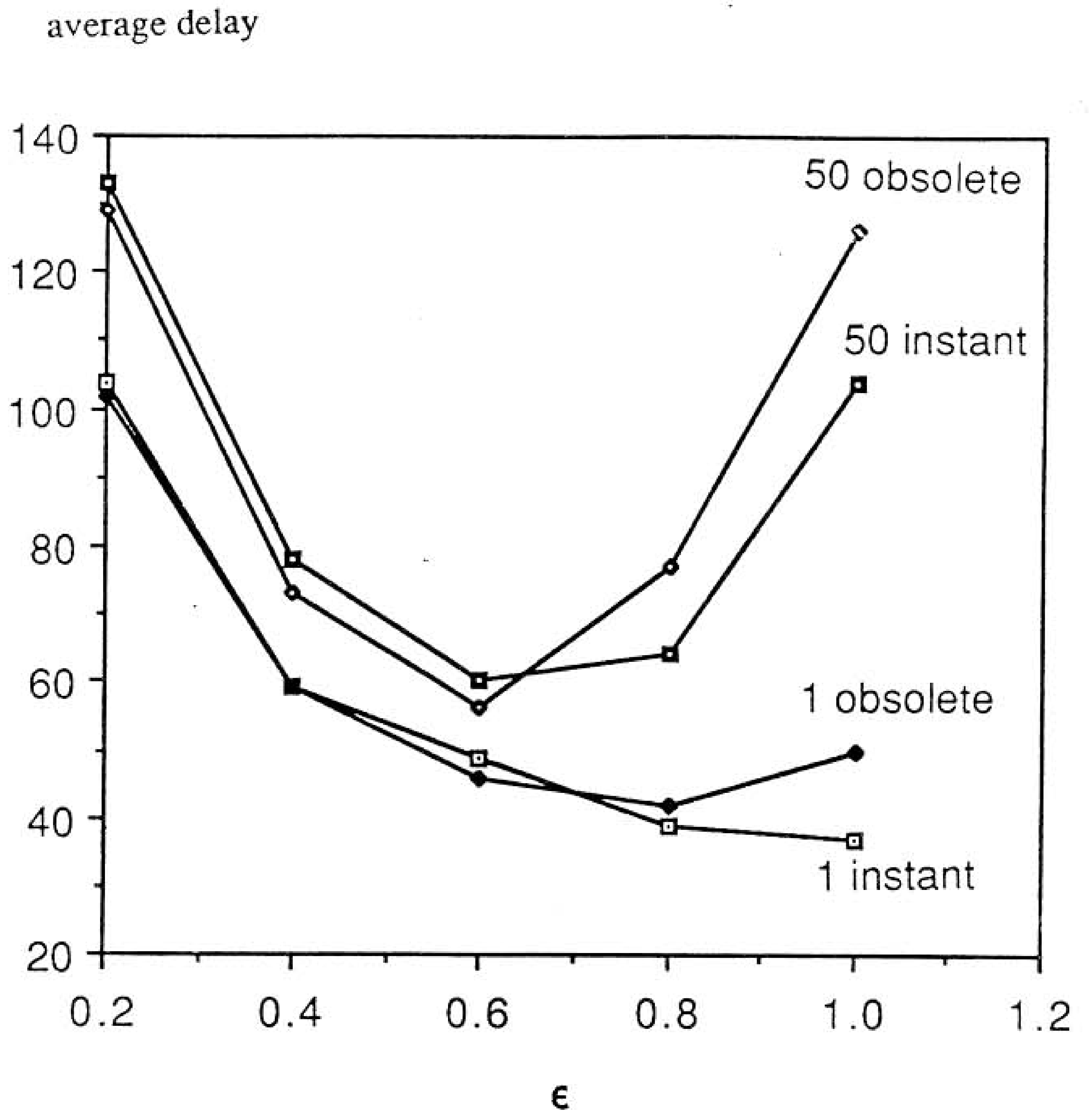

Figure 6.11: The average packet delay for the network of Figure 6.2 with $\gamma = 1/5$, $r = 1/50$, $\delta = 1/200$, for different values of the parameter ϵ, when we use as link length $l_{ij} = \epsilon * \frac{1 + N_{ij}}{{}_{ij}} + (1 - \epsilon) * \frac{1 + V_{ij}}{{}_{ij}}$.

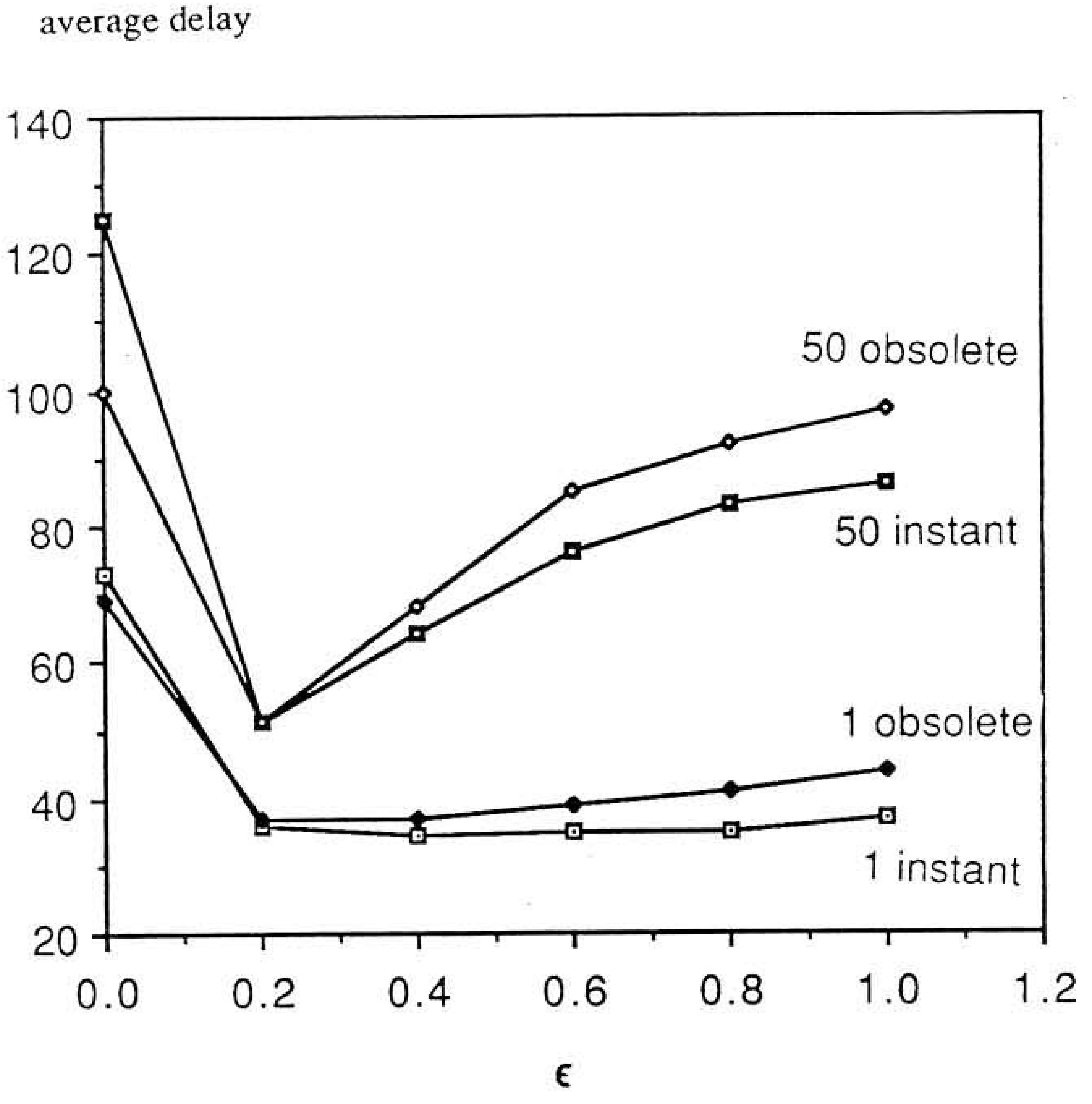

Figure 6.12: The average packet delay for the network of Figure 6.2 with $\gamma = 1/50$, $r = 1/5$, $\delta = 1/200$, for different values of the parameter ϵ, when we use as link length $l_{ij} = \epsilon * \frac{1 + N_{ij}}{{}_{ij}} + (1 - \epsilon) * \frac{1 + V_{ij}}{{}_{ij}}$.

frequently and therefore the network state changes more quickly. So, the state information that we use in the routing decisions is out-of-date.

6.9 Application to Integrated Services Networks

In this section, we make load sharing, routing and congestion control decisions in integrated services networks using stochastic learning automata.

The methology that we propose for learning automata-based load sharing, routing and congestion control decisions for new arriving virtual circuits (in connection-oriented ISN's) or new arriving packets (in connectionless-oriented ISN's) is similar to that of the previous section. So, we do not reiterate it here.

The only difference will be that the cost functions for each class in ISN's are different than those in virtual circuit networks. Therefore, the information that will be needed in order to calculate the lengths to destinations and the path lengths will be different than that for virtual circuit networks. For example, one class may use its blocking experience, while another class may use its packet delay to update its routing probability.

Chapter 7

Conclusions & Suggestions for Future Research

7.1 Conclusions

The major contribution of this dissertation is the introduction of a unified game-theoretic methodology for the multi-objective joint load sharing, routing and congestion control problem in distributed systems. And the introduction of stochastic learning automata algorithms for decentralized asynchronous computation of the solution.

We develop a novel mathematical approach, based on game theory, for the decentralized quasi-static and dynamic problem. After defining the joint problem, we model the distributed system on the path flow space using queueing and state space models. Then we develop three methodologies for both the quasi-static and the dynamic cases of the problem: i) *Team optimization methodology*, when the classes of jobs cooperate for the socially optimum, ii) *Nash game methodology*, when the classes of jobs compete among themselves and each class try to operate optimally for its own jobs and iii) *Stackelberg game methodology*, when some classes of jobs have more power than others, for example priority classes.

For each methodology, we formulate the problem as a *Nonlinear Programming or Optimal Control/Dynamic Programming, a Nonlinear Complementarity and a Variational Inequality problem.* We state conditions for existence/uniqueness of the solution and derive the optimality conditions for the quasi-static problem using

Karush-Kuhn-Tucker theorem, and for the dynamic problem using Pontryagin's maximum principle.

We apply the proposed methodologies to Datagram, Virtual Circuit and Integrated Services Networks and develop several new queueing models and performance measures, for each network type. We explicitly solve several examples and evaluate the system performance via simulation.

Finally, we introduce new classes of Stochastic Learning Automata algorithms and propose decentralized dynamic load sharing, routing and congestion control using Stochastic Learning Automata. Simulation is used to demonstrate improved system performance.

A variety of resource sharing problems arising in distributed systems may be formulated and solved using the proposed methodologies.

7.2 Suggestions for Future Research

Applications. We have presented several applications of the proposed methodologies and formulations that we have introduced in this dissertation. Obviously, we have not covered all possible applications. Thus, there is a huge research area to apply the proposed methodology. For example, by considering a specific network type (e.g. deterministic arrival and service distributions in ATM networks, threshold buffer management schemes, aging/deadline priorities, etc.), we may have different cost functions. Then selecting the appropriate scenario (cooperation, competition or hierarchy), we may formulate the problem as a team, Nash or Stackelberg game. Then we may choose either to solve the problem as a Nonlinear Programming, a Nonlinear Complementarity or a Variational Inequality Problem using appropriate algorithms.

Algorithms. We have developed several different formulations of the joint problem and suggested the use of iterative algorithms that solve the specific formulations. We have also introduced and tested via simulation one class of such decentralized, asynchronous dynamic algorithms, called stochastic learning automata.

Algorithms for solving cooperative or non-cooperative game problems are iterative algorithms that use first and possibly second derivatives. According to the iteration scheme, they can be classified as Gauss-Seidel, Successive Overrelaxation and Jacobi iteration algorithms. Instead of reiterating the existing bibliography on such algorithms, we rather refer to the original papers or books.

i) Nonlinear Programming algorithms: [152, 529, 339, 192, 30, 164, 165, 311, 387, 46],

ii) Optimal Control algorithms: [14, 292, 381, 415, 325, 131, 254, 412, 203, 440, 262, 301],

iii) Dynamic Programming algorithms: [220, 406, 45],

iv) Nash Games algorithms: [405, 429, 312, 175, 302, 110],

v) Stackelberg Games algorithms: [382, 54, 372, 371, 373, 442],

vi) Nonlinear Complementarity Problem algorithms: [113],

vii) Variational Inequalities algorithms: [198, 217].

Incentives. In this dissertation, we have assumed that the players either cooperate or compete for the resources. However, through the use of incentives, we can alter the scenario of the game and force the players to follow specific strategies.

Stochastic Discrete-Time. In chapter 5, we solved the dynamic deterministic optimal control problem, since we described the system state by the expected values of the stochastic processes. A direction for future research, is the solution of the stochastic problem either in continuous or discrete-time.

Hierarchical Games. In the Stackelberg game formulation, we considered two hierarchical levels, where at the upper level is the most powerful (e.g. higher priority) class of jobs and at the lower level (e.g. lower priority) is the less powerful class of jobs. One may extend these two levels to multiple hierarchical levels, where at each level there will be multiple classes. Then at each level the classes will play a Nash game, while classes among different levels will act as leaders and followers (Stackelberg game).

State Constraints. In chapter 5, we have introduced several possible constraints on the system state. However, we have not explicitly included them into the solution of the optimization problems. One may extend this research by explicitly solving the dynamic problem with state constraints.

Information Structure. In solving the dynamic problem, we have assumed that the current network state is known. One area for future research is to solve the dynamic problem with delayed information about the network state.

State Observation Structure. In solving the dynamic problem, we have assumed perfect information about the network state. Another area for future research is to solve the dynamic problem with imperfect state observation.

Reference List

[1] H.Z. Aashtiani and T.L. Magnanti. Equilibria on a congested transportation network. *SIAM J. ALG. DISC. METH.*, Vol. 2, No. 3, pp. 213-226, Sept. 1981.

[2] C.E. Agnew. On the optimality of adaptive routing algorithms. *Proc. National Telecommunications Conference*, pp. 1021-1025, IEEE 1974.

[3] C.E. Agnew. On quadratic adaptive routing. *Communications of ACM*, Vol. 19, No. 1, pp. 18-22, Jan. 1976.

[4] C.E. Agnew. Dynamic modeling and control of congestion-prone systems. *Operations Research*, Vol. 24, No. 3, pp. 400-419, May-June 1976.

[5] A.K. Agrawala and S.K. Tripathi. On the optimality of semidynamic routing schemes. *Information Processing Letters*, Vol. 13, No. 1, pp. 20-22, Oct. 1981.

[6] A.K. Agrawala, S.K. Tripathi, and G. Ricart. Adaptive routing using a virtual waiting time technique. *IEEE Transactions on Software Engineering*, Vol. SE-8, No. 1, pp. 76-81, Jan. 1981.

[7] M. Aicardi, F. Davoli, and R. Minciardi. Decentralized dynamic routing as a Markov chain optimization problem. *Proc. Conference on Decision and Control*, pp. 1514-1517, IEEE 1987.

[8] V.M. Alekseev, V.M. Tikhomirov, and S.V. Formin. *Optimal Control.* Consultants Bureau, 1987.

[9] C.D. Aliprantis, D.J. Brown, and O. Burkinshaw. *Existence and Optimality of Competitive Equilibria.* Springer-Verlag, 1989.

[10] A.O. Allen. *Probability, Statistics, and Queueing Theory with Computer Science Applications.* Academic Press, 1978.

[11] K.J. Arrow and F.W. Hahn. *General Competitive Analysis.* Holden-Day Inc., 1971.

[12] K.J. Arrow and M. Kurz. *Public Investment, The Rate of Return And Optimal Fiscal Policy.* The JohnsHopkins Press, 1970.

[13] H. Aso and M. Kimura. Absolutely expediency of learning automata. *Informations Sciences*, Vol. 17, pp. 91-112, 1979.

[14] M. Athans and P.L. Falb. *Optimal Control.* Mc Graw-Hill, 1966.

[15] R.C. Atkinson and G.H. Bower. *An Introduction to Mathematical Learning Theory.* J.Wiley & Sons, 1965.

[16] J.-P Aubin. *Mathematical Methods of Game and Economic Theory.* North Holland Publ. Co., 1979.

[17] R.J. Aumann. *Lectures on Game Theory.* Westview Press, Inc., 1989.

[18] A. Avritzer, M. Gerla, B.A.N. Ribeiro, J.W. Carlyle, and W.J. Karplus. The advantage of dynamic tuning in distributed asymmetric systems. *Proc. Infocom 90 Conference*, pp. 811-818, IEEE 1990.

[19] N. Baba. *New Topics in Learning Automata Theory and Applications.* Springer-Verlag, 1984.

[20] K. Bala, I. Cidon, and K. Sohraby. Congestion control for high speed packet switched networks. *Proc. Infocom 90 Conference*, pp. 520-526, IEEE 1990.

[21] S.A. Banawan and J. Zahorjan. Load sharing in heterogeneous queueing systems. *Proc. Infocom 89 Conference*, pp. 731-739, IEEE 1989.

[22] A.G. Barto and P. Anandan. Pattern-recognized stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 3, pp. 360-375, MAy/June 1985.

[23] A.G. Barto, R.S. Sutton, and C.W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 5, pp. 834-846, Sept./Oct. 1983.

[24] A.G. Barto, R.S. Sutton, and P.S. Brouwer. Associative search networks: a reinforcement learning associative memory. *Biological Cybernetics*, 40, pp. 201-211, 1081.

[25] T. Basar. Equilibrium strategies in dynamic games with multi-levels of hierarchy. *Automatica*, pp. 749-754, 1981.

[26] T. Basar and S. Li. Distributed computation of Nash equilibria in linear-quadratic stochastic differential games. *SIAM J. Control and Optimization*, Vol. 27, No. 3, pp. 563-578, May 1989.

[27] T. Basar and G.J. Olsder. *Dynamic Noncooperative Game Theory*. Academic Press, 1982.

[28] T. Basar and H. Selbuz. A new approach for derivation of closed-loop Stackelberg strategies. *Proc. Conference on Decision and Control*, pp. 1113-1118, IEEE 1978.

[29] T. Basar and H. Selbuz. Closed-loop Stackelberg strategies with applications in the optimal control of multilevel systems. *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 2, pp. 166-178, April 1979.

[30] M.S. Bazaraa and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. J.Wiley & Sons, 1979.

[31] R. Bellman. *Dynamic Programming*. Princeton Univ. Press, 1957.

[32] A.W. Berger. Performance analysis of a rate control throttle where tokens and jobs queue. *Proc. Infocom 90 Conference*, pp. 30-38, IEEE 1990.

[33] L.D. Berkovitz. *Optimal Control Theory*. Springer-Verlag, 1974.

[34] L.D. Berkovitz. Necessary conditions for optimal strategies in a class of differential games and control problems. *J. SIAM Control*, Vol. 5, No. 1, pp. 1-24, 1967.

[35] F. Bernabei, C. Calabro, and M. Listanti. A fully distributed routing control scheme in an ATM switch. *Proc. International Communications Conference*, pp. 766-770, IEEE 1990.

[36] D. Bertsekas and S. Shreve. *Stochastic Optimal Control : the Discrete Time Case*. Academic Press, 1978.

[37] D.P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Inc., 1987.

[38] D.P. Bertsekas. Dynamic models of shortest path routing algorithms for communication networks with multiple destinations. *Proc. Conference on Decision and Control*, pp. 127-133, IEEE 1979.

[39] D.P. Bertsekas. Optimal routing and flow control methods for communication networks. *Proc. 5th International Conference on Analysis and Optimization of Systems*, pp. 615-643, 1982.

[40] D.P. Bertsekas. A class of optimal routing algorithms for communication networks. *Proc. 5th Int. Conference on Computer Communications*, pp. 71-75, 1980.

[41] D.P. Bertsekas. Algorithms for nonlinear multicommodity network flow problems. *Proc. Int. Symposium on Systems Analysis and Optimization*, Springer-Verlag 1979.

[42] D.P. Bertsekas. Dynamic behavior of shortest path routing algorithms for communication networks. *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 1, pp. 60-74, Febr. 1982.

[43] D.P. Bertsekas and E.M. Gafni. Projected Newton methods and optimization of multicommodity flows. *IEEE Transactions on Automatic Control*, Vol. AC-28, No. 12, pp. 1090-1096, Dec. 1983.

[44] D.P. Bertsekas, E.M. Gafni, and R.G. Gallager. Second derivative algorithms for minimum delay distributed routing in networks. *IEEE Transactions on Communications*, Vol. COM-32, No. 8, pp. 911-919, Aug. 1984.

[45] D.P. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1987.

[46] D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., 1989.

[47] F.J. Beutler and D. Teneketzis. Optimal routing in queueing networks under imperfect information : stochastic dominance, thresholds and convexity. *ORSA/TIMS Joint National Meeting*, 1988.

[48] F.J. Beutler and D. Teneketzis. Routing in queueing networks under imperfect information: stochastic dominance and thresholds. *Stochastics and Stochastics Reports*, Vol. 26, pp. 81-100, 1989.

[49] K. Bharath-Kumar. Optimum end-to-end flow control in networks. *Proc. Int. Communications Conference*, pp. 23.3.1-23.3.6, IEEE 1980.

[50] K. Bharath-Kumar and J.M. Jaffe. A new approach to performance-oriented flow control. *IEEE Transactions on Communications*, Vol. COM-29, No. 4, pp. 427-435, July 1981.

[51] W.F. Bialas and M.H. Karwan. Multilevel optimization: a mathematical programming perspective. *Proc. Conference on Decision and Control*, pp. 761-765, IEEE 1980.

[52] W.F. Bialas and M.H. Karwan. On two-level optimization. *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 1, pp. 211-214, Febr. 1982.

[53] J.P.C. Blanc. A note on waiting times in systems with queues in parallel. *J. Applied Probability*, Vol. 24, pp. 540-546, 1987.

[54] J.W. Blankenship and J.E. Falk. Infinitely constrained optimization problems. *J. of Optimization Theory and Applications*, Vol. 19, No. 2, pp. 261-281, June 1976.

[55] A. Blaquiere, F. Gerard, and G. Leitmann. *Quantitative and Qualitative Games*. Academic Press, 1969.

[56] B.W. Boehm and R.L. Mobley. Adaptive routing techniques for distributed communications systems. *IEEE Transactions on Communications*, Vol. COM-17, No. 3, pp. 340-349, June 1969.

[57] R.K. Boel and J.H.van Schuppen. Distributed routing for load balancing. *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 210-221, Jan. 1989.

[58] V.G. Boltyanskii. *Mathematical Methods of Optimal Control*. Holt, Rinehart, Winston Inc., 1971.

[59] V.G. Boltyanskii. *Optimal Control of Discrete Systems*. J. Wiley & Sons, 1978.

[60] F. Bonomi. Performance analysis of some process-to-processor assignment algorithms for a UNIX multiprocessor system. *Distributed Processing*, M.H. Barton and E.L. Dagles and G.L. Reijns (eds.), pp. 491-504, Elsevier Science Publ., IFIP 1988.

[61] F. Bonomi. Adaptive optimal load balancing in a heterogeneous multiserver system with a central job scheduler. *8th Int, Conference on Distributed Computing Systems*, pp. 500-508, IEEE 1988.

[62] F. Bonomi. On job assignment for a parallel system of processor sharing queues. *IEEE Transactions on Computers*, Vol. 39, No.7, pp. 858-869, July 1990.

[63] F. Bonomi, P.J. Fleming, and P. Steinberg. An adaptive join-the biased-queue rule for load sharing on distributed computer systems. *Proc. 28th Conference on Decisions and Control*, pp. 2554-2559, IEEE 1988.

[64] F. Bonomi and A. Kumar. Optimality of weighted least squares load balancing. *Proc. 27th Conference on Decision and Control*, pp. 1480-1485, IEEE 1988.

[65] R.R. Boorstyn and A. Livne. A technique for adaptive routing in networks. *IEEE Transactions on Communications*, Vol. COM-29, No. 4, pp. 474-480, April 1981.

[66] K.C. Border. *Fixed Point Theorems with Applications to Economic and Game Theory*. Cambridge University Press, 1985.

[67] F. Borgonovo and E. Cadorin. Locally-optimal deflection routing in the Bidirectional Manhattan Network. *Proc. Infocom 90 Conference*, pp. 458-464, IEEE 1990.

[68] A.D. Bovopoulos. Resource allocation algorithms for packet switched networks. *Ph.D. Dissertation, Columbia University*, 1989.

[69] A.D. Bovopoulos. Resource allocation as a Nash game in a multiclass packet switched environment. *Computer Science Dept., Technical Report WUCS-89-18, pp. 1-9*, Washington University 1989.

[70] A.D. Bovopoulos and A.A. Lazar. Asynchronous algorithms for optimal flow control of BCMP networks. *Dept. of Computer Science, Washington University, WUCS-89-10*, pp. 1-22, 1989.

[71] A.D. Bovopoulos and A.A. Lazar. Decentralized network flow control. *Proc. Int. Computer Communications Conference*, pp. 139-143, 1988.

[72] A.D. Bovopoulos and A.A. Lazar. Optimal load balancing algorithms for Jacksonian networks with acknowledgement delay. *Proc. Computer Networking Symposium*, pp. 144-151, IEEE 1988.

[73] A.D. Bovopoulos and A.A. Lazar. Optimal routing and flow control of a network of parallel processors with individual buffers. *Proc. 23rd Annual Allerton Conference on Communications, Control, and Computing*, pp. 564-573, 1985.

[74] A.D. Bovopoulos and A.A. Lazar. Load balancing algorithms for Jacksonian networks with acknowledgement delays. *Proc. Infocom '89 Conference*, pp. 749-757, IEEE 1989.

[75] A.D. Bovopoulos and A.A. Lazar. Decentralized algorithms for optimal flow control. *25th Annual Allerton Conference on Communications, Control, and Computing*, pp. 979-988, 1987.

[76] J. Bracken and J.T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, Vol. 21, No. 1, pp. 37-44, 1973.

[77] S.C. Bruell and G. Balbo. *Computational Algorithms for Closed Queueing Networks*. North Holland, 1980.

[78] R. Bryant and R.A. Finkel. A stable distributed scheduling algorithm. *Proc. Second Int. Conf. on Distributed Computing Systems*, pp.314-323, April 1981.

[79] A.E.Jr. Bryson and Y. C. Ho. *Applied Optimal Control*. Hemisphere Publ. Co, 1975.

[80] R.R. Bush and F. Mosteller. *Stochastic Models and Learning*. J.Wiley & Sons, 1955.

[81] J.P. Buzen and P.P.S. Chen. Optimal load balancing in memory hierarchies. *Information Processing 74*, pp. 271-275, North Holland Publ. Co. 1974.

[82] D.H. Cansever. Decentralized algorithms for flow control in networks. *Proc. 25th Conference on Decision and Control*, pp. 2107-2112, IEEE 1986.

[83] D.G. Cantor and M. Gerla. Optimal routing in a packet-switched computer network. *IEEE Transactions on Computers*, Vol. C-23, No. 10, pp. 1062-1069, Oct. 1974.

[84] G. Casalino, F. Davoli, R. Minciardi, and R. Zoppoli. On the structure of decentralized dynamic routing strategies. *Proc. Conference on Decision and Control*, pp. 472-476, IEEE 1983.

[85] T.L. Casavant and J.G. Kuhl. Analysis of three dynamic distributed load-balancing strategies with varying global information requirements. *Proc. Int. Conf. on Distributed Computing Systems*, pp. 185-192, IEEE 1987.

[86] T.L. Casavant and J.G. Kuhl. Effects of response and stability on scheduling in distributed computing systems. *IEEE Transactions on Software Engineering*, Vol. 14, No. 11, pp. 1578-1588, Nov. 1988.

[87] J.H. Case. *Economics and the Competitive Process*. New York University Press, 1979.

[88] J.H. Case. Toward a theory of many player differential games. *SIAM J. Control*, Vol. 7, No. 2, pp. 179-197, May 1969.

[89] C.G. Cassandras, V.V. Abide, and D. Towsley. Distributed routing with on-line marginal delay estimation. *IEEE Transactions on Communications*, Vol. 38, No. 3, pp. 348-359, March 1990.

[90] C.G. Cassandras, M.H. Kallmes, and D. Towsley. Optimal routing and flow control in networks with real-time traffic. *Proc. Infocom 89 Conference*, pp. 784-791, IEEE 1989.

[91] D.A. Castanon and N.R.Jr. Sandel. On hierarchical Stackelberg optimization problems. *Proc. Conference on Decision and Control*, pp. 104-107, IEEE 1977.

[92] B. Chandrasekaran and D.W.C. Shen. On expedience and convergence in variable-structure automata. *IEEE Transactions on Systems Science and Cybernetics*, Vol. SSC-4, No. 1, pp. 52-60, March 1968.

[93] C.-J. Chang and J.-F. Chang. The effect of idle server first random routing in the behavior of a finite queue. *IEEE Transactions on Communications*, Vol. COM-35, No. 5, pp. 496-502, May 1987.

[94] F. Chang and L. Wu. An optimal adaptive routing algorithm. *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 8, pp. 690-700, Aug. 1986.

[95] T.-S. Chang and P.B. Luh. Derivation of necessary and sufficient conditions for single-stage Stackelberg games via the inducible region concept. *IEEE Transactions on Automatic Control*, Vol. AC-29, No. 1, pp. 63-66, Jan. 1984.

[96] C.I. Chen and J.B.Jr. Cruz. Stackelberg solution for two-person games with biased information patterns. *IEEE Transactions on Automatic Control*, Vol. AC-17, No. 6, pp. 791-797, Dec. 1972.

[97] M.S. Chen and J.S. Meditch. A distributed adaptive routing algorithm. *Proc. International Communications Conference*, pp. B5.3.1-B5.3.9, IEEE 1983.

[98] L.A. Chenault. On the uniqueness of Nash equilibria. *Economic Letters*, Vol. 20, pp. 203-205, 1986.

[99] T.C.K. Chou and J.A. Abraham. Load balancing in distributed systems. *IEEE Trans. on Software Eng., Vol SE-8, No4*, pp. 401-412, July 1982.

[100] T.C.K. Chou and J.A. Abraham. Distributed control of computer systems. *IEEE Trans. on Computers, Vol C-35, No 6*, pp.564-567, June 1986.

[101] W. Chou, A.W. Bragg, and A.A. Nilsson. The need for adaptive routing in the chaotic and unbalanced traffic environment. *IEEE Transactions on Communications*, Vol. COM-29, No. 4, pp. 481-490, April 1981.

[102] T.-C. Chow and W.H. Kohler. Models for dynamic load balancing in a heterogeneous multiple processor system. *IEEE Trans. on Computers, Vol C-28, No 5*, pp. 354-361, May 1979.

[103] Y.C. Chow and W.H. Kohler. Dynamic load balancing in homogeneous two-processor distributed systems. *Int. Symposium on Computer Performance Modeling, Measurement and Evaluation*, pp. 39-52, August 1977.

[104] M.S. Chrystall and P. Mars. Adaptive routing in computer communication networks using learning automata. *Proc. of IEEE Nat. Telecomm. Conf.*, pp. A3.2.1-7, 1981.

[105] P.H.N. Chu, R.R. Boorstyn, and A. Kershenbaum. A simulation study of a dynamic routing scheme. *Proc. National Telecommunications Conference*, pp. A3.4.1-A3.4.11, IEEE 1981.

[106] W.W. Chu and M.Y. Shen. A hierarchical routing and flow control policy (HRFC) for packet switched networks. *Proc. Computer Performance Conference*, ACM 1977.

[107] W.W. Chu and M.Y. Shen. A hierarchical routing and flow control policy (HRFC) for packet switched networks. *IEEE Transactions on Computers*, Vol. C-29, No. 11, pp. 971-977, Nov. 1980.

[108] B. Ciciani, D.M. Dias, and P.S. Yu. Load sharing in hybrid distributed-centralized database systems. *Proc. 8th Int. Conference on Distributed Computing Systems*, pp. 274-281, IEEE 1988.

[109] G.M. Clark. Use of Polya distributions in approximate solutions to M/M/s queues. *Communications of the ACM*, Vol. 24, No. 4, pp. 206-217, April 1981.

[110] G. Cohen. Nash equilibria: gradient and decomposition algorithms. *Large Scale Systems*, Vol. 12, pp. 173-184, 1987.

[111] A.E. Conway and N.D. Georgana. *Queueing Networks - Exact Computational Algorithms.* The MIT Press, 1990.

[112] R.B. Cooper. *Introduction to Queueing Theory.* North Holland, 2nd ed. 1981.

[113] R.W. Cottle, F. Giannesi, and J.-L. Lions. *Variational Inequalities and Complementarity Problems: Theory and Applications.* J. Wiley & Sons, 1980.

[114] P.-J. Courtois and P. Semal. An algorithm for the optimization of nonbifurcated flows in computer communication networks. *Performance Evaluation*, pp. 139-152, 1981.

[115] P.J. Courtois. *Decomposability: Queueing and Computer System Applications*. Academic Press, 1977.

[116] D.R. Cox and W.L. Smith. *Queues*. Chapman and Hall, 1961.

[117] J.B.Jr. Cruz. Leader-follower strategies for multilevel systems. *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 2, pp. 244-254, April 1978.

[118] S. Dafermos. Traffic equilibrium and variational inequalities. *Transportation Science*, Vol. 14, No. 1, pp. 42-54, Febr. 1980.

[119] S. Dafermos. An iterative scheme for variational inequalities. *Mathematical Programming*, Vol. 26, pp. 40-47, 1983.

[120] S.C. Dafermos and F.T. Sparrow. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards B.*, Mathematical Sciences, Vol. 73B, No. 2, pp.91-118, April-June 1969.

[121] B. Daneshrad and S.D. Morgera. Application of stochastic automaton theory for routing in a packet-switched network. *Proc. MILCOM*, pp. 205-209, IEEE 1989.

[122] J.M. Danskin. *The Theory of Max-Min*. Springer-Verlag, 1967.

[123] J.P. Dauer and W. Stadler. A survey of vector optimization in infinite-dimensional spaces, part 2. *J. of Optimization Theory and Applications*, Vol. 51, No. 2, pp. 205-241, Nov. 1986.

[124] E. de Souza e Silva and M. Gerla. Load balancing in distributed systems with multiple classes and site constraints. *Performance '84, E. Gelenbe (ed.)*, pp. 17-33, North Holland 1984.

[125] M. Decina and T. Toniatti. On bandwidth allocation to bursty virtual connections in ATM networks. *Proc. Int. Communications Conference*, pp. 844-850, IEEE 1990.

[126] M. Decina, T. Toniatti, P. Vaccari, and L. Verri. Bandwidth assignment and virtual call blocking in ATM networks. *Proc. Infocom 90 Conference*, pp. 881-888, IEEE 1990.

[127] R.L. Disney and P.C. Kiessler. *Traffic Processes in Queueing Networks: A Markov Renewal Approach.* The Johns Hopkins University Press, 1987.

[128] C. Douligeris and R. Mazumdar. A game theoretic approach to flow control in an integrated environment with two classes of users. *Proc. IEEE Computer Networking Symposium, pp. 214-221*, 1988.

[129] C. Douligeris and R. Mazumdar. More on Pareto optimal flow control. *Proc. 26th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1181-1190, 1988.

[130] C. Douligeris and R. Mazumdar. On Pareto optimal flow control in a multiclass environment. *Proc. 25th Annual Allerton Conference on Communication, Control, and Computing*, pp. 989-997, 1987.

[131] P. Dyer and S.R. McReynolds. *The Computation and Theory of Optimal Control.* Academic Press, 1970.

[132] D.L. Eager, E.D. Lazowska, and J. Zahorjan. A comparison of receiver-initiated and sender-initiated adaptive load sharing. *ACM Sigmetrics Conf. on Measurement and Modeling of Computer Systems*, pp. 1-3, ACM 1985.

[133] D.L. Eager, E.D. Lazowska, and J. Zahorjan. The limiting performance benefits of migrating active processes for load sharing. *Proc. SIGMETRICS*, pp. 63-72, ACM 1988.

[134] D.L. Eager, E.D. Lazowska, and J. Zahorjan. Adaptive load sharing in homogeneous distributed systems. *IEEE Trans. on Software Eng., Vol SE-12, No 5*, pp. 662-675, May 1986.

[135] A.E.Jr. Eckberg, D.T. Luan, and D.M. Lucantoni. Meeting the challenge: Congestion and flow control strategies for Broadband information transport. *Proc. GLOBECOM 1989*, pp. 1769-1773, IEEE 1989.

[136] A.A. Economides, P.A. Ioannou, and J.A. Silvester. Decentralized adaptive routing for virtual circuit networks using stochastic learning automata. *Proc. of IEEE Infocom 88 Conference*, pp. 613-622, IEEE 1988.

[137] A.A. Economides, P.A. Ioannou, and J.A. Silvester. Adaptive routing and congestion control for window flow controlled virtual circuit networks. *Proc. 27th Annual Allerton Conference on Communications, Control, and Computing*, pp. 849-857, Sept. 1989.

[138] A.A. Economides and J.A. Silvester. Priority load sharing: an approach using Stackelberg games. *USC Technical Report CENG 89-39*, 1989.

[139] A.A. Economides and J.A. Silvester. Routing games. *USC Technical Report CENG 89-38*, 1989.

[140] A.A. Economides and J.A. Silvester. A game theory approach to cooperative and non-cooperative routing problems. *Proc. IEEE International Telecommunications Symposium*, Brazil, Sept. 3-6, 1990.

[141] A.A. Economides and J.A. Silvester. Optimal routing in a network with unreliable links. *Proc. of IEEE Computer Networking Symposium*, pp. 288-297, IEEE 1988.

[142] K. Efe and B. Groselj. Minimizing control overheads in adaptive load sharing. *Proc. 9th Int. Conference on Distributed Computing Systems*, pp. 307-315, IEEE 1989.

[143] Y.M. El-Fattah. *Learning Systems: Decisions, Simulation, and Control.* Springer-Verlag, 1978.

[144] A. Ephremides. Extension of an adaptive distributed routing algorithm to mixed media networks. *IEEE Transactions on Communications*, Vol. COM-26, No. 8, pp. 1262-1266, Aug. 1978.

[145] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 4, Aug. 1980.

[146] D. Ferguson, Y. Yemini, and C. Nikolaou. Microeconomic approach for load balancing in distributed computer systems. *Proc. 8th Int. Conference on Distributed Computing Systems*, pp. 491-499, IEEE 1988.

[147] D.F. Ferguson, C. Nikolaou, and Y. Yemini. An economy for flow control in computer networks. *Proc. Infocom '89 Conference*, pp. 110-118, IEEE 1989.

[148] D. Ferrari. A study of load indices for load balancing schemes. *Workload Characterization of Computer Systems*, G. Serazzi (ed.), pp. 91-99, Elsevier Science Publ. 1986.

[149] D. Ferrari, G. Serazzi, and A. Zeigner. *Measurement and Tuning of Computer Systems.* Prentice Hall, 1983.

[150] D. Ferrari and S. Zhou. An empirical investigation of load indices for load balancing algorithms. *Performance 87*, P.-J. Courtois and G. Latouche (eds), pp. 515-528, Elsevier Science Publ. 1988.

[151] D. Ferrari and S. Zhou. A load index for dynamic load balancing. *Proc. Fall Joint Computer Conference*, pp. 684-690, IEEE 1986.

[152] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. J. Wiley & Sons, 1968.

[153] J. Filipiak. *Modelling and Control of Dynamic Flows in Communication Networks*. Springer-Verlag, 1988.

[154] J. Filipiak. M-architecture: A structural model of traffic management and control in broadband ISDNs. *IEEE Communication Magazine*, pp. 25-31, May 1989.

[155] J. Filipiak. Analysis and synthesis of routing and flow control in a nonstationary environment. *Proc. of International Conference on Computer Communications*, pp. 446-451, 1985.

[156] J. Filipiak. Optimization of dynamic flows in networks of hierarchical structure. *Problems of Control and Information Theory*, Vol. 11, No. 3, pp. 155-165, 1982.

[157] J. Filipiak. Unloading of congestion in deterministic queueing networks. *Optimal Control Applications & Methods*, Vol. 2, pp. 23-45, 1981.

[158] J. Filipiak. Optimal control of store-and-forward networks. *Optimal Control Applications & Methods*, Vol. 3, pp. 155-176, 1982.

[159] J. Filipiak. Dynamic routing in queueing systems with a multiple service facility. *Operations Research*, Vol. 32, No. 5, pp. 1163-1180, 1984.

[160] J. Filipiak. Structured systems analysis methodology for desigh of an ATM network architecture. *Journal On Selected Areas In Communications*, Vol. JSAC-7, No. 8, pp. 1263-1273, Oct. 1989.

[161] L. Flatto and H.P. McKean. Two queues in parallel. *Communications on Pure and Applied Mathematics*, Vol. xxx, pp. 255-263, 1977.

[162] W.H. Fleming and R.W. Rishel. *Deterministic and Stochastic Optimal Control.* Springer-Verlag, 1975.

[163] Y.A. Flerov. Some classes of multi-input automata. *Journal of Cybernetics*, Vol. 2, No. 3, pp. 112-122, 1972.

[164] R. Fletcher. *Practical Methods of Optimization; Vol. 1: Unconstraint Optimization.* J.Wiley & Sons, 1980.

[165] R. Fletcher. *Practical Methods of Optimization; Vol. 2: Constraint Optimization.* J.Wiley & Sons, 1981.

[166] L.R. Ford and D.R. Fulkerson. *Flows in Networks.* Princeton University Press, 1962.

[167] G. Foschini and J. Salz. A basic dynamic routing problem and diffusion. *IEEE Transactions on Communications*, Vol. COM-26, No. 3, pp. 320-327, March 1978.

[168] G.J. Foschini. On heavy traffic diffusion analysis and dynamic routing in packet switched networks. *Computer Performance*, K.M.Chandy and M.Reiser (eds.), North Holland Publ. Co., pp. 499-513, 1977.

[169] H. Frank. Dynamic communication networks. *IEEE Transactions on Communication Technology*, Vol. COM-15, No. 2, pp. 156-163, April 1967.

[170] H. Frank and M.T. El-Bardai. Dynamic communication networks with capacity constraints. *IEEE Transactions on Communication Technology*, Vol. COM-17, No. 4, pp. 432-437, Aug. 1969.

[171] H. Frank and I.T. Frisch. *Communication, Transmission, and Transportation Networks.* Addison-Wesley, 1971.

[172] L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method : an approach to store-and-forward communication network design. *Networks, 3*, pp. 97-133, 1973.

[173] A. Friedman. *Differential Games.* J. Wiley & Sons, 1971.

[174] J.W. Friedman. *Oligopoly and the theory of games.* North Holland, 1977.

[175] D. Gabay and H. Moulin. On the uniqueness and stability of Nash-equilibria in noncooperative games. *Applied Stochastic Control in Econometrics and Management Science*, A.Bensoussan et al. (eds.), pp. 271-293, North Holland 1980.

[176] E.M. Gafni. and D.P. Bertsekas. Path assignment for virtual circuit routing. *Proc. ACM SIGCOMM 1983 Conference*, pp. 21-25, ACM 1983.

[177] E.M. Gafni and D.P. Bertsekas. Asymptotic optimality of shortest path routing algorithms. *IEEE Trans. on Information Theory*, Vol. IT-33, No. 1, pp. 83-90, Jan. 1987.

[178] R.G. Gallager. Distributed network optimization algorithms. *Proc. International Communications Conference*, pp. 43.2.1-43.2.2, IEEE 1979.

[179] R.G. Gallager. A minimum delay routing using distributed computation. *IEEE Trans. on Communications*, Vol. COM-25, No. 1, Jan. 1977.

[180] R.G. Gallager and S.J. Golestaani. Flow control and routing algorithms for data networks. *Proc. International Computer Communications Conference*, pp. 779-784, 1980.

[181] C. Gao, J.W.S. Liu, and M. Railey. Load balancing algorithms in homogeneous distributed systems. *Proc. Int. Conference on Parallel Processing*, pp. 302-306, IEEE 1984.

[182] B. Gavish and S.L. Hantler. An algorithm for optimal route selection in sna networks. *IEEE Transactions on Communications*, Vol. COM-31, No. 4, pp. 1154-1161, 1983.

[183] B. Gavish and I. Neuman. A system for routing and capacity assignment in computer communication networks. *IEEE Transactions on Communications*, Vol. COM-37, No. 4, pp. 360-366, Apr. 1989.

[184] E. Gelenbe and I. Mitrani. *Analysis and Synthesis of Computer Systems.* Academic Press, 1980.

[185] E. Gelenbe and G. Pujolle. *Introduction to Queueing Networks.* J.Wiley & Sons, 1987.

[186] M. Gerla. Packet, circuit and virtual circuit switching. *in W. Chou (ed.) "Computer Communications vol II : Systems and Applications"*, pp. 222-267, Prentice Hall 1985.

[187] M. Gerla, H.W. Chan, and J.R. Boisson De Marca. Routing, flow control and fairness in computer networks. *Proc. of International Communications Conference*, pp. 1272-1275, IEEE 1984.

[188] M. Gerla and P.O. Nilsson. Routing and flow control interplay in computer networks. *Proc. International Conference on Computer Communications*, pp. 84-89, 1980.

[189] A. Gersht. Analytical model of dynamic routing in virtual circuit packet switching network. *Proc. International Conference on Computer Communications*, pp.76-80, IEEE 1982.

[190] A. Gersht and K.J. Lee. Virtual-circuit load control in fast packet-switched broadband networks. *Prc. GLOBECOM*, pp. 214-220, IEEE 1988.

[191] A. Gersht and K.L. Lee. A congestion control framework for ATM networks. *Proc. Infocom'89 Conference*, pp. 701-710, IEEE 1989.

[192] P.E. Gill and W. Murray. *Numerical Methods for Constrained Optimization.* Academic Press, 1974.

[193] D.W. Glazer and C. Tropper. A new metric for dynamic routing algorithms. *IEEE Transactions on Communications*, Vol. 38, No. 3, pp. 360-367, March 1990.

[194] R.M. Glorioso and F.C. Colon. Cybernetic control of computer networks. *Modeling and Simulation Vol. 5, No 2, Proc. Fifth Annual Pittsburg Conf.*, pp.819-824, April 1974.

[195] R.M. Glorioso, G.R. Grueneich, and J.C. Dunn. Self organization and adaptive routing for communication networks. *EASCON '69 Record*, pp.243-250, 1969.

[196] R.M. Glorioso, G.R. Grueneich, and D. McElroy. Adaptive routing in a large communication network. *Proc. 9th Symposium on Adaptive Processes*, pp. XV.5.1-XV.5.4, 1970.

[197] R.M. Glorioso and F.C.C. Osorio. *Engineering Intelligent Systems: Concepts, Theory, and Applications.* Digital Press, 1980.

[198] R. Glowinski, J.T. Lions, and R. Tremolieres. *Numerical Analysis of Variational Inequalities.* North-Holland, 1981.

[199] S.J. Golestani. Congestion-free communication in broadband packet networks. *Proc. Int. Communications Conference*, pp. 489-494, IEEE 1990.

[200] S.J. Golestani. Congestion-free transmission of real-time traffic in packet networks. *Proc. Infocom 90 Conference*, pp. 527-536, IEEE 1990.

[201] P.M. Gopal, B.K. Kadaba, and G. Wieber. Load distribution in packet-switched networks. *Proc. IEEE International Conference on Communications*, pp. 980-986, IEEE 1987.

[202] D. Gross and C.M. Harris. *Fundamentals of Queueing Theory.* J.Wiley & Sons, 2nd ed. 1985.

[203] W.A. Gruver and E. Sachs. *Algorithmic Methods in Optimal Control.* Pitman Advanced Publ., 1980.

[204] W. Guth and B. Kalkofen. *Unique Solutions for Strategic Games.* Springer-Verlag, 1989.

[205] A. Hac and X. Jin. Dynamic load balancing in a distributed system using a decentralized algorithm. *Proc. Int. Conference on Distributed Computing Systems*, pp. 170-177, IEEE 1987.

[206] A. Hac and X. Jin. Dynamic load balancing in a distributed system using a sender-initiated algorithm. *Proc. Local Computer Networks*, pp. 172-180, IEEE 1988.

[207] A. Hac and T.J. Johnson. A study of dynamic load balancing in a distributed system. *Proc. SIGGCOM*, pp. 348-356, ACM 1986.

[208] E.L. Hahne. Dynamic routing in an unreliable manufacturing network with limited storage. *Lab. for Information and Decision Systems, MIT*, LIDS-TH-1063, OSP No. 87049, Febr. 1981.

[209] E.L. Hahne and R.G. Gallager. Round robin scheduling for fair flow control in data communication networks. *Proc. International Communications Conference*, pp. 103-107, IEEE 1986.

[210] B. Hajek and R.G. Ogier. Optimal dynamic routing in communication networks with continuous traffic. *Proc. Conference on Decision and Control*, pp. 369-374, IEEE 1982.

[211] B. Hajek and R.G. Ogier. Optimal dynamic routing in communication networks with continuous traffic. *Networks*, Vol. 14, pp. 457-487, 1984.

[212] S. Halfin. The shortest queue problem. *J. Applied Probability*, Vol. 22, pp. 865-875, 1985.

[213] J.C. Harsanyi and R. Selten. *A General Theory of Equilibrium Selection in Games.* The MIT Press, 1988.

[214] J.F. Hayes. *Modeling and Analysis of Computer Communication Networks.* Plenum Press, 1984.

[215] M.R. Hestenes. *Calculus of Variations and Optimal Control Theory.* J. Wiley & Sons, 1966.

[216] M. Hirano and N. Watanabe. Characteristics of a cell multiplexer for bursty ATM traffic. *Proc. Int. Communications Conference*, pp. 399-403, IEEE 1989.

[217] I. Hlavacek, J. Haslinger, J. Necas, and J. Lovisek. *Solution of Variational Inequalities in Mechanics.* Springer-Verlag, 1988.

[218] Y.-C. Ho. Team decision theory and information structure. *Proceedings of the IEEE*, Vol. 68, No, 6, pp. 644-654, June 1969.

[219] T.-C. Hou and D.M. Lucantoni. Performance analysis of an integrated video/data transport mechanism with built-in congestion control. *Proc. GLOBECOM*, pp. 231-238, IEEE 1988.

[220] R.A. Howard. *Dynamic Programming and Markov Processes.* The MIT Press, 1960.

[221] M.-T. Hsiao. Optimal decentralized flow control in computer communication networks. *Ph.D. Dissertation, Columbia University*, 1986.

[222] M.-T. Hsiao and A.A. Lazar. Optimal flow control of multi-class queueing networks with decentralized information. *Proc. IEEE Infocom 87*, pp. 652-661, IEEE 1987.

[223] M.-T.T. Hsiao and A. Lazar. A game theoretic approach to decentralized flow control of Markovian queueing networks. *Proc. Performance '87*, P.-J. Courtois and G. Latouche (eds.), Elsevier Sc. Publ., pp. 55-73, 1988.

[224] M.-T.T. Hsiao and A.A. Lazar. Optimal decentralized flow control of Markovian queueing networks with multiple controllers, part I: the team decision problem. *Data Communications Performance Evaluation*, pp. 357-372, 1987.

[225] M.-T.T. Hsiao and A.A. Lazar. Optimal flow control of multiclass queueing networks with partial information. *IEEE Transactions on Automatic Control*, Vol. 35, No. 7, pp. 855-860, July 1990.

[226] C.-Y.H. Hsu and J.W.-S. Liu. Dynamic load balancing algorithms in homogeneous distributed systems. *Proc.*, pp.216-223, IEEE 1986.

[227] J.Y. Hui. Resource allocation for broadband networks. *Journal On Selected Areas In Communications*, Vol. JSAC-6, pp. 1598-1608, 1988.

[228] P.A. Humblet, S.R. Soloway, and B. Steinka. Algorithms for data communication networks-part 2. *Codex*, pp. 1-23, Jun. 1986.

[229] B.M. Ibrahim and A.K. Elhakeem. Combined adaptive routing/capacity assignment in nested movable boundary frame integrated services networks. *Proc. GLOBECOM*, pp. 1867-1873, IEEE 1989.

[230] M.A. Iqbal, J.H. Saltz, and S.H. Bokhari. A comparative analysis of static and dynamic load balancing strategies. *Proc. Int. Conference on Parallel Processing*, pp. 1040-1047, IEEE 1986.

[231] R. Isaacs. *Differential Games.* J. Wiley & Sons, 1965.

[232] V. Jacobson. Congestion avoidance and control. *Proc. SIGCOMM*, pp. 314-329, ACM 1988.

[233] J.M. Jaffe. A decentralized "optimal", multiple-use, flow control algorithm. *Proc. International Computer Communications Conference*, pp. 839-843, 1980.

[234] J.M. Jaffe and A. Segall. Threshold design for dynamic routing. *Proc. IEEE International Communications Conference*, pp. 90-92, IEEE 1986.

[235] N.K. Jaiswal. *Priority Queues.* Academic Press, 1968.

[236] W. Jianhua. *The Theory of Games.* Oxford University Press, 1988.

[237] M.I. Kamien and N.L. Schwartz. Sufficient conditions in optimal control theory. *Journal Economic Theory*, Vol. 3, pp. 207-214, 1971.

[238] T. Kamitake and T. Suda. Evaluation of an admission control scheme for an ATM network considering fluctuations in cell loss rate. *Proc. GLOBECOM*, pp. 1774-1780, IEEE 1989.

[239] F. Kamoun. A drop and throttle flow control policy for computer networks. *IEEE Transactions on Communications*, Vol. COM-29, No. 4, pp. 444452, April 1981.

[240] F. Kamoun, A. Belguith, and J.L. Grange. Congestion control with a buffer management strategy based on traffic priorities. *Proc. Int. Conference on Computer Communications*, pp. 845-850, 1990.

[241] F. Kamoun and L. Kleinrock. Stochastic performance evaluation of hierarchical routing for large networks. *Computer Networks*, 3, pp. 337-353, 1979.

[242] E.L. Kaplan. *Mathematical Programming and Games.* J. Wiley & Sons, 1982.

[243] S. Karamardian. The complementarity problem. *Mathematical Programming*, Vol. 2, pp. 107-129, 1972.

[244] S. Karamardian. The nonlinear complementarity problem with applications, part 1. *J. of Optimization Theory and Applications*, Vol. 4, No. 2, pp. 87-98, 1969.

[245] S. Karamardian. The nonlinear complementarity problem with applications, part 2. *J. of Optimization Theory and Applications*, Vol. 4, No. 3, pp. 167-181, 1969.

[246] S. Karamardian. Generalized complementarity problem. *J. of Optimization Theory and Applicaions*, Vol. 8, No. 3, pp. 161-168, 1971.

[247] M.G.H. Katevenis. Fast switching and fair control of congested flow in broadband networks. *IEEE Journal on Selected Areas in Communications*, Vol. SAC-5, No. 8, pp. 1315-1326, Oct. 1987.

[248] S. Katz and I. Rubin. Session admission control for a multi-access communication channel. *Proc. Infocom '87 Conference*, pp. 635-642, IEEE 1987.

[249] F.P. Kelly. *Reversibility and Stochastic Networks.* J.Wiley & Sons, 1979.

[250] A. Khanna and J. Zinky. The revised ARPANET routing metric. *Proc. Communication Architectures and Protocols*, pp. 45-56, ACM 1989.

[251] A.Y. Khintchine. *Mathematical Methods in the Theory of Queues.* Hafner Publ. Co., 2nd ed., 1969.

[252] B.G. Kim and D. Towsley. Dynamic flow control protocols for packet-switching multiplexers serving real-time multipacket messages. *IEEE Transactions on Communications*, Vol. COM-34, No. 4, pp. 348-356, April 1986.

[253] D. Kinderlehrer and G. Stampacchia. *An Introduction to Variational Inequalities and their Applications.* Academic Press, 1980.

[254] D.E. Kirk. *Optimal Control Theory: an Introduction.* Prentice Hall, 1970.

[255] L. Kleinrock. *Queueing Systems, Vol. 1: Theory.* J. Wiley & Sons, 1975.

[256] L. Kleinrock. *Queueing Systems, Vol. 2: Applications.* J. Wiley & Sons, 1976.

[257] L. Kleinrock and W. Korfhage. Collecting unused processing capacity: an analysis of transient distributed systems. *Proc. 9th Int. Conference on Distributed Computing Systems*, pp. 482-489, IEEE 1989.

[258] L. Kleinrock and C.W. Tseng. Flow control based on limiting permit generation rates. *Proc. Int. Conference on Computer Communications*, pp. 785-790, 1980.

[259] A.H. Klopf. *The Hedonistic Neuron*. Hemisphere Publ. Co., 1982.

[260] C. Knessl, B.J. Matkowsky, Z. Schuss, and C. Tier. Two parallel queues with dynamic routing. *IEEE Transactions on Communications*, Vol. COM-34, No. 12, pp. 1170-1175, Dec. 1986.

[261] C. Knessl, B.J. Matkowsky, Z. Schuss, and C. Tier. Two parallel M/G/1 queues where arrivals join the system with the smaller buffer content. *IEEE Transactions on Communications*, Vol. COM-35, No. 11, pp. 1153-1158, Nov. 1987.

[262] G. Knowles. *An Introduction to Applied Optimal Control*. Academic Press, 1981.

[263] H. Kobayashi. *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*. Addison Wesley, 1978.

[264] H. Kobayashi and M. Gerla. Optimal routing in closed queueing networks. *ACM Transactions on Computer Systems*, Vol. 1, No. 4, pp. 294-310, Nov. 1983.

[265] N.N. Krasovskii and A.I. Subbotin. *Game-Theoretical Control Problems*. Springer-Verlag, 1988.

[266] I.A. Krass and S.M. Hammoudeh. *The Theory of Positional Games with Applications in Economics*. Academic Press, 1981.

[267] V.L. Kreps. Finite n-person non-cooperative games with unique equilibrium points. *International Journal of Game Theory*, Vol. 10, No. 3/4, pp. 125-129, 1981.

[268] K.R. Krishnan. Performance benefits of state-dependent routing of telephone traffic. *Proc. International Communications Conference*, pp. 1314-1318, IEEE 1990.

[269] K.R. Krishnan. Joining the right queue: a state dependent decision rule. *IEEE Transactions on Automatic Control*, Vol. 35, No. 1, pp. 104-107, Jan. 1990.

[270] H. Kroner. Comparative performance study of space priority mechanisms for ATM networks. *Proc. Infocom 90 Conference*, pp. 1136-1143, IEEE 1990.

[271] P. Krueger and M. Livny. The diverse objectives of distributed scheduling policies. *Proc. Int. Conference on Distributed Computing Systems*, pp. 242-249, IEEE 1987.

[272] A. Kumar. Adaptive load balancing in a multi-processor system with a central job scheduler. *Computer Performance and Reliability*, G. Iazeolla and P.J. Courtois and O.J. Boxma (eds), pp. 173-188, Elsevier Science Publ. 1988.

[273] A. Kumar. Adaptive load control of a central processor in a distributed system with a star topology. *IEEE Transactions on Computers*, Vol. 38. No. 11, pp. 1502-1512, Nov. 1989.

[274] P. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control.* Prentice Hall, 1986.

[275] P.R. Kumar and J. Walrand. Individually optimal routing in parallel systems. *J. Applied Probability*, Vol. 22, pp. 989-995, 1985.

[276] J.F. Kurose and R. Chipalkatti. Load sharing in soft real-time distributed computer systems. *IEEE Transactions on Computers*, Vol. C-36, No. 8, pp. 993-1000, Aug. 1987.

[277] J.F. Kurose and R. Simha. Second derivative algorithms for optimal resource allocation in distributed computer systems. *Proc. Int. Conference on Distributed Computing Systems*, pp. 56-63, IEEE 1987.

[278] J.F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, Vol. 38, No. 5, pp. 705-717, May 1989.

[279] J.F. Kurose and S. Simha. A distributed algorithm for optimum static load balancing in distributed computer systems. *Proc. Infocom 86 Conference*, pp. 458-467, IEEE 1986.

[280] J.F. Kurose, S. Singh, and R. Chipalkatti. A study of quasi-dynamic load sharing in soft real-time distributed computer systems. *Proc. Real Time Systems Symposium*, IEEE 1986.

[281] H.J. Kushner and H. Huang. Averaging methods for the asymptotic analysis of learning and adaptive systems, with small adjustment rate. *SIAM J. Control and Optimization*, Vol. 19, pp. 635-650, 1981.

[282] S. Lakshmivarahan. *Learning Algorithms Theory and Applications*. Springer Verlag, 1981.

[283] S. Lakshmivarahan and M.A.L. Thathachar. Absolutely expedient learning algorithms for stochastic automata. *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 81-286, May 1973.

[284] S.S. Lam. Congestion control of packet communication networks by input buffer limits - a simulation study. *IEEE Transactions on Computers*, Vol. C-30, No. 10, pp. 733-742, Oct. 1981.

[285] S.S Lam and Y.C.L. Lien. Modeling and analysis of flow controlled packet switching networks. *Proc. 7th Data Communications Symposium*, pp.98-107, IEEE 1981.

[286] S.S. Lam and Y.L. Lien. Optimal routing in networks with flow-controlled virtual channels. *Proc. Computer Networks Performance Evaluation*, pp.38-46, ACM 1982.

[287] S.S. Lam and M. Reiser. Congestion control of store-and-forward networks by input buffer limits-an analysis. *IEEE Transactions on Communications*, Vol. COM-27, No. 1, pp. 127-134, Jan. 1979.

[288] S.S. Lavenberg. *Computer Performance Modeling Handbook.* Academic Press, 1983.

[289] A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, 1982.

[290] A.A. Lazar and M.-T. Hsiao. Network and user optimal flow control with decentralized information. *Proc. IEEE Infocom 86*, pp. 468-477, IEEE 1986.

[291] E. Lazowska, J. Zahorjan, G.S. Graham, and K.C. Sevcik. *Quantitative System Performance: Computer System Analysis Using Queueing Network Models.* Prentice-Hall, Inc., 1984.

[292] E.B. Lee and L. Markus. *Foundations of Optimal Control Theory.* J. Wiley & Sons, 1967.

[293] K.J. Lee and D. Towsley. A comparison of priority-based decentralized load balancing policies. *Proc. SIGMETRICS*, pp. 70-77, ACM 1986.

[294] K.J. Lee and D. Towsley. An asymptotic analysis of a threshold load balancing policy. *Proc. Infocom 89 Conference*, pp. 740-748, IEEE 1989.

[295] K.J. Lee, D. Towsley, and M. Choi. Distributed algorithms for minimum delay routing with constraints in communication networks. *Proc. Infocom Conference*, pp. 188-199, IEEE 1987.

[296] A. Leff, P.S. Yu, and Y.-H. Lee. Adaptive transaction routing in a heterogeneous database environment. *Proc. 9th Int. Conference on Distributed Computing Systems*, pp. 406-413, IEEE 1989.

[297] G. Leitmann. *Cooperative and Non-cooperative Many Player Differential Games*. Springer-Verlag, 1974.

[298] G. Leitmann. *The Calculus of Variations and Optimal Control: An Introduction*. Plenum Press, 1981.

[299] W.E. Leland and T.J. Ott. Unix process behavior and load balancing among loosely-coupled computers. *Teletraffic Analysis and Computer Performance Evaluation*, O.J. Boxma, J.W. Cohen, H.C. Tijms (eds.), pp. 191-208, Elsevier Science Publ. 1986.

[300] W.E. Leland and T.J. Ott. Load-balancing and process behavior. *Proc.*, pp. 54-69, ACM 1986.

[301] F.L. Lewis. *Optimal Control*. J. Wiley & Sons, 1986.

[302] S. Li and T. Basar. Distributed algorithms for the computation of noncooperative equilibria. *Automatica*, Vol. 23, No. 4, pp. 523-533, 1987.

[303] H.-C. Lin, J.R. Yee, and C.S. Raghavendra. Optimal joint load balancing and routing in message switched computer networks. *Proc. Infocom 88 Conference*, pp. 274-281, IEEE 1988.

[304] Y.-S. Lin and J.R. Yee. A distributed routing algorithm for virtual circuit data networks. *Proc. of IEEE infocom 89*, pp. 200-207, 1989.

[305] J.L. Lions and G. Stampacchia. Variational inequalities. *Communications on Pure and Applied Mathematics*, Vol. xx, pp. 493-519, 1967.

[306] H.T. Liu and J. Silvester. A two-mode dynamic algorithm (TDMA) for load balancing in distributed systems. *Second Annual Parallel Processing Symposium*, pp. 189-215, 1988.

[307] H.T. Liu and J. Silvester. An approximate performance model for load-dependent interactive queues with application to load balancing in distributed systems. *Proc. Infocom 88*, pp. 956-965, IEEE 1888.

[308] M. Livny and M. Melman. Load balancing in homogeneous broadcast distributed systems. *Proc. ACM Computer Network Performance Symposium*, pp.47-55, April 1982.

[309] H. Lu and M.J. Carey. Load-balanced task allocation in locally distributed computer systems. *Proc. Int. Conference on Parallel Processing*, pp. 1037-1039, IEEE 1986.

[310] R.D. Luce and H. Raiffa. *Games and Decisions*. J. Wiley & Sons, 1957.

[311] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publ. Co., 2nd ed., 1984.

[312] D.G. Luenberger. Complete stability of noncooperative games. *J. of Optimization Theory and Applications*, Vol. 25, No. 4, pp. 485-505, Aug. 1978.

[313] P.B. Luh, S.-C. Chang, and T.-S. Chang. Solutions and properties of multistage Stackelberg games. *Automatica*, Vol. 20, No. 2, pp. 251-256, 1984.

[314] M.H. MacDougall. *Simulating Computer Systems*. The MIT Press, 1987.

[315] J. Macki and A. Strauss. *Introduction to Optimal Control*. Springer-Verlag, 1982.

[316] E.A. MacNair and C.H. Sauer. *Elements of Practical Performance Modeling*. Prentice Hall, 1985.

[317] B. Maglaris, R. Boorstyn, S. Panwar, and T. Spirtos. Routing in burst-switched voice/data integrated networks. *Proc. Infocom 87 Conference*, pp. 162-169, IEEE 1987.

[318] B. Maglaris, R. Boorstyn, S. Panwar, and T. Spirtos. Routing of voice and data in burst-switched networks. *Transactions on Communications*, Vol. 38, No. 6, pp. 889-897, June 1990.

[319] B.S. Maglaris. An optimal local policy for two-level adaptive routing in computer networks. *Proc. of IEEE Infocom 84 Conf.*, 1984.

[320] O.L. Mangasarian. Sufficient conditions for the optimal control of nonlinear systems. *Journal SIAM Control*, Vol. 4, No. 1, pp. 139-152, 1966.

[321] L.G. Mason. Learning automata and telecommunications switching. *Third Yale Workshop on Applications of Adaptive Systems Theory*, pp.120-128, 1983.

[322] L.G. Mason. Equilibrium flows, routing patterns and algorithms for store-and-forward networks. *Large Scale Systems*, Vol. 8, pp. 187-209, 1985.

[323] J. Matsumoto and H. Mori. Flow control in packet-switched networks by gradual restrictions of virtual calls. *IEEE Transactions on Communications*, Vol. COM-29, No. 4, pp. 466-473, April 1981.

[324] N.F. Maxemchuk and M.El Zarki. Routing and flow control in high-speed wide-area networks. *Proceedings of the IEEE*, Vol. 78, No. 1, pp. 204-221, January 1990.

[325] I. McCausland. *Introduction to Optimal Control.* J. Wiley & Sons, 1969.

[326] J.M. McQuillan, I. Richer, and E.C. Rosen. The new routing algorithm for the ARPANET. *IEEE Transactions on Communications*, Vol. COJM-28, No. 5, pp. 711-719, May 1980.

[327] J.S. Meditch. On the state-space approach in modelingdata-communication networks. *Proc. 15th Annual Allerton Conference on Communication, Control, and Computing*, 1977.

[328] A. Mehlmann. *Applied Differential Games.* Plenum Press, 1988.

[329] R. Mirchandaney and J.A. Stankovic. Using stochastic learning automata for job scheduling in distributed processing systems. *Journal of Parallel, and Distributed Computing, 3*, pp. 527-552, 1986.

[330] R. Mirchandaney, D. Towsley, and J.A. Stankovic. Adaptive load sharing in heterogeneous systems. *Proc. 9th Int. Conference on Distributed Computing Systems*, pp. 298-306, IEEE 1989.

[331] R. Mirchandaney, D. Towsley, and J.A. Stankovic. Analysis of the effects of delays on load sharing. *IEEE Transactions on Computers*, Vol. 38, No. 11, pp. 1513-1525, Nov. 1989.

[332] I. Mitrani. *Modeling of Computer and Communication Systems.* Cambridge University Press, 1987.

[333] S.P. Morgan. Window flow control on a trunked bute-stream virtual circuit. *IEEE Tr. on Communications*, Vol. 36, No. 7, pp. 816-825, July 1988.

[334] P. Morse. *Queues, Inventories, and Maintenance.* J. Wiley & Sons, 1958.

[335] F.H. Moss and A. Segall. An optimal control approach to dynamic routing in networks. *IEEE Tr. on Automatic Control*, Vol.AC -27, No. 2, pp. 329-339, Apr. 1982.

[336] H. Moulin. *Game Theory for the Social Science*. New York University Press, 2nd ed., 1986.

[337] K.H. Muralidhar and M.K. Sundareshan. A hierarchical scheme for multi-objective adaptive routing in large communication networks. *Proceedings of the IEEE*, Vol. 71, No. 12, pp. 1461-1463, Dec. 1983.

[338] K.H. Muralidhar and M.K. Sundareshan. Combined routing and flow control in computer communication networks: a two-level adaptive scheme. *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 1, pp. 15-25, Jan. 1987.

[339] W. Murray. *Numerical Methods for Unconstrained Optimization*. Academic Press, 1972.

[340] M.W. Mutka and M. Livny. Scheduling remore processing capacity in a workstation-processor bank network. *Proc. Int. Conference on Distributed Computing Systems*, pp. 2-9, IEEE 1987.

[341] S. Narasimhan, H. Pirkul, and P. De. Route selection in backbone data communication networks. *Computer Networks and ISDN Systems*, Vol. 15, pp. 121-133, 1988.

[342] K. Narendra and M.A.L. Thathacher. *Learning Automata: An Introduction*. Prentice Hall, 1989.

[343] K.S. Narendra and M.A.L. Thathachar. On the behavior of a learning automaton in a changing environment with application to telephone traffic routing. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-10, No. 5, May 1980.

[344] K.S. Narendra and M.A.L. Thathachar. Learning automata : A survey. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-4, No. 4, pp. 323-334, July 1974.

[345] K.S. Narendra and R.M. Wheeler. Routing in communication networks - a case study of learning in large scale systems. *Large Scale Systems 8*, pp. 211-222, 1985.

[346] K.S. Narendra, E.A. Wright, and L.G. Mason. Application of learning automata to telephone traffic routing and control. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-7, No.11, pp. 785-792, Nov. 1977.

[347] J. Nash. Non-cooperative games. *Annals of Mathematics*, Vol. 54, No. 2, pp. 286-295, Sept. 1951.

[348] J.F.Jr. Nash. Equilibrium points in n-person games. *Proc. N.A.S. Mathematics*, Vol. 36, pp. 48-49, 1950.

[349] O.V. Nedzelnitsky. Learning algorithms in data communication networks. *Proc. Third Yale Workshop on Applications of Adaptive Systems Theory*, 1983.

[350] O.V.Jr. Nedzelnitsky and K.S. Narendra. Nonstationary models of learning automata routing in data communication networks. *IEEE Tr. on Systems, Man and Cybernetics*, Vol. SMC-17, No. 6, pp. 1004-1015, Nov./Dec. 1987.

[351] F. Neelamkavil. *Computer Simulation and Modelling.* J.Wiley & Sons, 1986.

[352] L.M. Ni and K. Hwang. Optimal load balancing strategies for a multiple processor system. *Proc. Int. Conference on Parallel Processing*, pp. 352-357, IEEE 1981.

[353] L.M. Ni and K. Hwang. Optimal load balancing in a multiple processor system with many job classes. *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 5, pp. 491-496, May IEEE 1985.

[354] L.M. Ni, C.-W. Xu, and T.B. Gendreau. A distributed drafting algorithm for load balancing. *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 10, pp. 1153-1161, Oct. 1985.

[355] M.F. Norman. *Markov Processes and Learning Models.* Academic Press, 1972.

[356] M.F. Norman. Markovian learning processes. *SIAM Review*, Vol. 16, No. 2, pp. 143-162, April 1974.

[357] M.F. Norman. A central limit theorem for Markov processes that move by small steps. *The Annals of Probability*, Vol. 2, No. 6, pp. 1065-1074, 1974.

[358] R.G. Ogier. Minimum-delay routing in continuous-time dynamic networks with piecewise-constraint capacities. *Networks*, Vol. 18, pp. 303-318, 1988.

[359] H. Ohnishi, T. Okada, and K. Noguchi. Flow control schemes and delay/loss tradeoff in ATM networks. *IEEE J. Selected Areas in Communications*, Vol. 6, No. 9, pp. 1609-1616, IEEE 1988.

[360] H. Okazaki and M. Schwartz. Preliminary results on routing in hybrid link systems. *Proc. International Communications Conference*, pp. 1527-1532, IEEE 1988.

[361] K. Okuguchi. Expectations and stability in oligopoly models. *Springer-Verlag*, 1976.

[362] G.J. Olsder and R. Suri. Time-optimal control of parts routing in a manufacturing system with failure-prone machines. *Proc. of IEEE 19th Conf. on Decision and Control*, pp.722-727, 1980.

[363] B.J. Oommen. Ergodic learning automata capable of incorporating a priori information. *IEEE Transactions on Systems, Man, and Cybernetics*, Vo. SMC-17, No. 4, pp. 717-723, July/Aug. 1987.

[364] B.J. Oommen. Absorbing and ergodic discretized two-action learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-16, No. 2, pp. 282-293, March/April 1986.

[365] B.J. Oommen and J.P.R. Christensen. ϵ-optimal discretized linear reward-penalty learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-18, No. 3, pp. 451-458, May/June 1988.

[366] B.J. Oommen and D.C.Y. Ma. Deterministic learning automata to the equipartitioning problem. *IEEE Transactions on Computers*, Vol. 37, No. 1, pp. 2-13, Jan. 1988.

[367] B.J. Oommen and M.A.L. Thathachar. Multiaction learning automata possessing ergodicity of the mean. *Information Sciences*, Vol. 35, pp. 183-198, 1985.

[368] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, 1970.

[369] G. Owen. *Game Theory.* Academic Press, 2nd ed., 1982.

[370] K.S. Narendra P. Mars and M. Crystall. Learning automata control of computer communications networks. *Third Yale Workshop on Applications of Adaptive Systems Theory*, pp. 114-119, 1983.

[371] G.P. Papavassilopoulos. Algorithms for static Stackelberg games with linear costs and polyhedra constraints. *Proc. Conference on Decisions and Control*, pp. 647-652, IEEE 1982.

[372] G.P. Papavassilopoulos. Algorithms for leader-follower games. *Proc. 18th Annual Allerton Conference on Communication, Control, and Computing*, pp. 851-859, 1980.

[373] G.P. Papavassilopoulos. Solution of some stochastic quadratic Nash and leader-follower games. *SIAM J. Control and Optimization*, Vol. 19, No. 4, pp. 651-666, Sept. 1981.

[374] G.P. Papavassilopoulos. On the linear-quadratic-Gaussian Nash game with one-step delay observation sharing pattern. *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 5, pp. 1054-1071, Oct. 1982.

[375] G.P. Papavassilopoulos and J.B.Jr. Cruz. Sufficient conditions for Stackelberg and Nash strategies with memory. *J. of Optimization Theory and Applications*, Vol. 31, No. 2, pp. 233-260, June 1980.

[376] G.P. Papavassilopoulos and J.B.Jr. Cruz. Nonclassical control problems and Stackelberg games. *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 2, pp. 155-165, April 1979.

[377] F. Pavlidou. A variable reduction routing algorithm. *Proc. GLOBECOM*, pp. 2129-2132, IEEE 1987.

[378] M.C. Pennotti and M. Schartz. Congestion control in store and forward tandem links. *IEEE Tr. on Communications*, Vol. COM-23, No. 12, pp. 1434-1443, Dec. 1975.

[379] D.W. Peterson and J.H. Zalkind. A review of direct sufficient conditions in optimal control theory. *International Journal Control*, Vol. 28, No. 4, pp. 589-610, 1978.

[380] D.W. Petr and V.S. Frost. Optimal packet discarding: an ATM-oriented analysis model and initial results. *Proc. Infocom 90 Conference*, pp. 537-542, IEEE 1990.

[381] J.B. Plant. *Some Iterative Solutions in Optimal Control.* The MIT Press, 1968.

[382] E. Polak and D.Q. Mayne. An algorithm for optimization problems with functional inequality constraints. *IEEE Transactions on Automatic Control*, Vol. AC-21, No. 2, pp. 184-193, April 1976.

[383] B.T. Polyak. Convergence and convergence rate of iterative stochastic algorithms, I. the general case. *Avtomatika i Telemekhanika*, No. 12, pp. 83-94, Dec. 1976.

[384] B.T. Polyak. Convergence and convergence rate of iterative stochastic algorithms, II. the linear case. *Avtomatika i Telemekhanika*, No. 4, pp. 101-107, April 1977.

[385] J.-P. Ponssard. *Competitive Strategies.* North Holland Publ. Co., 1981.

[386] A.S. Poznyak. Investigation of the convergence of algorithms for the functioning of learning stochastic automata. *Avtomatika i Telemekhanika*, No. 1, pp. 88-103, Jan. 1975.

[387] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes.* Cambridge University Press, 1986.

[388] G. Pujolle, D. Seret, D. Dromard, and E. Horlait. *Integrated Digital Communications Networks, Vol.1.* J. Wiley & Sons, 1988.

[389] G. Pujolle, D. Seret, D. Dromard, and E. Horlait. *Integrated Digital Communications Networks, Vol.2.* J. Wiley & Sons, 1988.

[390] S. Pulidas, D. Towsley, and J.A. Stankovic. Imbedding gradient estimators in load balancing algorithms. *8th Int. Conf. on Distributed Computing Systems*, pp. 482-490, IEEE 1988.

[391] J. Quirk and R. Saposnik. *Introduction to General Equilibrium Theory and welfare Economics.* McGraw-Hill Inc., 1968.

[392] K.K. Ramakrishnan and R. Jain. A binary feedback scheme for congestion avoidance in computer networks. *Transactions on Computer Systems*, Vol. 8, No. 2, pp. 158-181, May 1990.

[393] C.V. Ramamoorthy and W.-T. Tsai. An adaptive hierarchical algorithm. *Proc. COMPSAC*, pp. 93-104, IEEE 1983.

[394] G. Ramamurthy and R.S. Dighe. Integration of high speed continuous stream data traffic in a broadband packet network. *Proc. Infocom 89 Conference*, pp. 1063-1071, IEEE 1989.

[395] G. Ramamurthy and R.S. Dighe. Distributed source control: a network access control for integrated broadband packet networks. *Proc. Infocom 90 Conference*, pp. 896-907, IEEE 1990.

[396] A. Rapoport. *N-person Game Theory.* The University of Michigan Press, 1970.

[397] D.A. Reed and C.Kim. Packet routing algorithms for integrated switching networks. *Proc. SIGMETRICS*, pp. 7-15, ACM 1987.

[398] K.L. Rider. A simple approximation to the average queue size in the time-dependent M/M/1 queue. *Journal of the Association for Computing Machinery*, Vol. 23, No. 2, pp. 361-367, April 1976.

[399] J. Riordan. *Stochastic Service Systems.* J. Wiley & Sons, 1962.

[400] Z. Rosberg. Optimal dispatching to parallel heterogeneous servers in light traffic. *Proc. 28th Conference on Decision and Control*, pp. 1539-1543, IEEE 1989.

[401] Z. Rosberg. Deterministic routing to buffered channels. *IEEE Transactions on Communications*, Vol. COM-34, No. 5, pp. 504-507, May 1986.

[402] Z. Rosberg and A.M. Makowski. Optimal routing to parallel heterogeneous servers- small arrival rates. *IEEE Transactions on Automatic Control*, Vol. 35, No. 7, pp. 789-796, July 1990.

[403] J.B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, Vol. 33, No. 3, pp. 520-534, July 1965.

[404] J. Rosenmuller. *The Theory of Games and Markets.* North-Holland Publ. Co., 1981.

[405] J. Rosenmuller. On a generalization of the Lemke-Howson algorithm to noncooperative n-person games. *SIAM J. Applied Mathematics*, Vol. 21, No. 1, pp. 73-79, July 1971.

[406] S. Ross. *Introduction to Stochastic Dynamic Programming.* Academic Press, 1983.

[407] M.H. Rothkopf and S.S. Oren. A closure approximation for the nonstationary M/M/s queue. *Management Science*, Vol. 25, No. 6, pp. 522-534, June 1979.

[408] R.Y. Rubinstein. *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks.* J.Wiley & Sons, 1986.

[409] H. Rudin. On routing and delta routing: a taxonomy and performance comparison of techniques for packet-switched networks. *IEEE Transactions on Communications*, Vol. COM-24, No. 1, pp. 43-59, Jan. 1978.

[410] H. Rudin and H. Muller. On routing and flow control. *Flow control in computer networks*, J.-L. Grange and M. Gien (eds.), pp. 241-255, North-Holland Publ. Co. 1979.

[411] H. Rudin and H. Muller. More on routing and flow control. *Proc. National Telecommunications Conference*, pp. 34.5.1-34.5.9, IEEE 1979.

[412] D.L. Russell. *Mathematics of Finite-Dimensional Control Systems.* Marcel Dekker Inc., 11979.

[413] S. Saad and M. Schwartz. Input buffer limiting mechanisms for congestion control. *Proc. Int. Communications Conference*, pp. 23.1.1-23.1.5, IEEE 1980.

[414] T.L. Saaty. *Elements of Queueing Theory with Applications.* Mc Graw-Hill, 1961.

[415] A.P. Sage. *Optimum Systems Control.* Prentice Hall, 1968.

[416] V.R. Saksena. Analysis of routing issues in the design of packet networks. *IFAC Large Scale Systems*, pp. 803-808, 1986.

[417] N.S.Jr. Sandell. On open-loop and closed-loop Nash strategies. *IEEE Transactions on Automatic Control*, pp. 435-436, Aug. 1974.

[418] B.A. Sanders. An asynchronous, distributed flow control algorithm for rate allocation in computer networks. *IEEE Transactions on Computers*, Vol. 37, No. 7, pp. 779-787, July 1988.

[419] B.A. Sanders. An incentive compatible flow control algorithm for rate allocation in computer networks. *IEEE Transactions on Computers*, Vol. 37, No. 9, pp. 1067-1072, Sept. 1988.

[420] B.A. Sanders. A private good/public good decomposition for optimal flow control of an M/M/1 queue. *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 11, pp. 1143-1145, Nov. 1985.

[421] P.E. Sarachik. A dynamic alternate route strategy fo traffic networks. *Proc. 21st Conference on Decision and Control*, pp. 120-124, IEEE 1982.

[422] P.E. Sarachik. Decentralized dynamic clearing of congested multi-destination networks. *Proc. 19th Annual Allerton Conference on Communication, Control, and Computing*, pp. 797-803, 1981.

[423] P.E. Sarachik. An effective local dynamic strategy to clear congested multi-destination networks. *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 2, pp. 510-513, Apr. 1982.

[424] P.E. Sarachik and U. Ozguner. On decentralized dynamic routing for congested traffic networks. *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 6, pp. 1233-1238, Dec. 1982.

[425] I.G. Sarma, R.K. Ragade, and U.R. Prasadi. Necessary conditions for optimal strategies in a class of noncooperative n-person differential games. *SIAM J. Control*, Vol. 7, No. 4, pp. 637-644, Nov. 1969.

[426] G. Sasaki and B. Hajek. Optimal dynamic routing in single commodity networks by iterative methods. *IEEE Tr. on Communications*, Vol. COM-35, No. 11, pp. 1199-1205, Nov. 1987.

[427] C.H. Sauer and K. M. Chandy. *Computer Systems Performance Modeling.* Prentice Hall, 1981.

[428] C.H. Sauer and E.A. MacNair. *Simulation of Computer Communication Systems.* Prentice Hall, 1983.

[429] H. Scarf. *The Computation of Economic Equilibria.* Yale University Press, 1973.

[430] H. Schulzrinne, J.F. Kurose, and D. Towsley. Congestion control for real-time traffic in high-speed networks. *Proc. Infocom 90 Conference*, pp. 543-550, IEEE 1990.

[431] M. Schwartz. *Computer Communication Network Design and Analysis.* Prentice Hall, 1977.

[432] M. Schwartz. *Telecommunication Networks: Protocols, Modeling and Analysis.* Addison-Wesley, 1987.

[433] M. Schwartz and C.K. Cheung. The gradient projection algorithm for multiple routing in message-switched networks. *IEEE Transactions on Communications*, pp. 449-456, April 1976.

[434] M. Schwartz and S. Saad. Analysis of congestion control techniques in computer communication networks. *Flow Control in Computer Networks*, J.-L. Grange and M. Gien (eds.), pp. 113-130, North-Holland Publ. Co. 1979.

[435] M. Schwartz and T.E. Stern. Routing technique used in computer communication networks. *IEEE Tr. on Communications*, Vol. COM-28, No. 4, pp. 539-552, Apr. 1980.

[436] M. Schwartz and T.-K. Yum. Distributed routing in computer communication networks. *Proc. Conference on Decision and Control*, pp. 600-603, IEEE 1982.

[437] A. Segall. The modeling of adaptive routing in data communication networks. *IEEE Tr. on Communications*, Vol. COM-25, No. 1, pp. 85-95, Jan. 1977.

[438] J. Seidler. *Principles of Computer Communication Network Design.* Ellis Horwood Lim., 1983.

[439] A. Seierstad and K. Sydsaeter. Sufficient conditions in optimal control theory. *International Economic Review*, Vol. 18, No. 2, pp. 367-391, June 1977.

[440] S.P. Sethi and G.L. Thompson. *Optimal Control Theory: Applications to Management Science*. Martinus Nijhoff Pu., 1981.

[441] S. Shenker and A. Weinrib. The optimal control of heterogeneous queueing systems: A paradigm for load-sharing and routing. *IEEE Transactions on Computers*, Vol. 38, No 12, pp. 1724-1735, Dec. 1989.

[442] K. Shimizu and E. Aiyoshi. A new computational method for Stackelberg and min-max problems by use of a penalty method. *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 2, pp. 460-466, April 1981.

[443] M. Shubik. *Games for Society, Business and War: Towards a Theory of Gaming*. Elsevier Scientific Publ. Co., 1975.

[444] M. Shubik. *Game Theory in the Social Sciences: Concepts and Solutions*. The MIT Press, 1982.

[445] M. Shubik. *A Game-Theoretic Approach to Political Economy*. The MIT Press, 1984.

[446] M. Sidi, W.-Z. Liu, and I. Gopal. Congestion control through input rate regulation. *Proc. GLOBECOM 1989*, pp. 1764-1768, IEEE 1989.

[447] M. Simaan. Stackelberg optimization of two-level systems. *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 554-557, July 1977.

[448] M. Simaan and J.B.Jr. Cruz. On the Stackelberg strategy in nonzero-sum games. *J. of Optimization Theory and Applications*, Vol. 11, No. 5, pp. 533-555, 1973.

[449] M. Simaan and J.B.Jr. Cruz. Additional aspects of the Stackelberg strategy in nonzero-sum games. *J. of Optimization Theory and Applications*, Vol. 11, No. 6. pp. 613-626, 1973.

[450] R. Simha and J.F. Kurose. Stochastic approximation schemes for a load balancing problem. *27th Annual Allerton Conference on Communication, Control, and Computing*, pp. 839-848, 1989.

[451] R. Simha and J.F. Kurose. Relative reward strength algorithms for learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 2, pp. 388-398, March/April 1989.

[452] M.J. Smith. Existence, uniqueness and stability of traffic equilibria. *Transportation Research*, Vol. 13B, pp. 295-304, 1979.

[453] P.R. Srikantakumar. Application of learning theory to communication networks control. *Third Yale Workshop on Applications of Adaptive Systems Theory*, pp. 135-141, 1983.

[454] P.R. Srikantakumar. Adaptive routing in large communication networks : Probabilistic study. *Proc. IEEE Conf. on Decision and Control*, pp. 398-401, 1981.

[455] P.R. Srikantakumar. Application of learning theory to communication networks control. *Proc. IEEE Cybernetics and Society Conf.*, pp. 555-559, 1983.

[456] P.R. Srikantakumar and K.S. Narendra. A learning model for routing in telephone networks. *SIAM J. Control and Optimization*, vol-20, no 1, Jan. 1982.

[457] K. Sriram. Dynamic bandwidth allocation and congestion control schemes for voice and data multiplexing in wideband packet technology. *Proc. Infocom'90 Conference*, pp. 1003-1009, IEEE 1990.

[458] W. Stadler. A survey of multicriteria optimization or the vector maximum problem. *J. of Optimization Theory and Applications*, Vol. 29, No. 1, pp. 1-52, Sept. 1979.

[459] H. Stalford and G. Leitmann. Sufficiency conditions for n-person differential games. *Topics in Differential Games*, A. Blaquiere (ed.), pp. 345-376, North Holland 1973.

[460] J.A. Stankovic. The analysis of a decentralized control algorithm for job scheduling utilizing Bayesian decision theory. *Proc. Int. Conference on Parallel Processing*, pp. 333-340, IEEE 1981.

[461] J.A. Stankovic. Simulations of three adaptive, decentralized controlled, job scheduling algorithms. *Computer Networks*, Vol. 8, pp. 199-217, 1984.

[462] J.A. Stankovic. An application of Bayesian decision theory to decentralized control of job scheduling. *IEEE Transactions on Computers*, Vol. C-34, No. 2, pp. 117-130, Feb. 1985.

[463] J.A. Stankovic. Stability and distributed scheduling algorithms. *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 10, pp. 1141-1152, 1985.

[464] A.W. Starr and Y.C. Ho. Nonzero-sum differential games. *J. on Optimization Theory and Applications*, Vol. 3, No. 3, pp. 184-206, 1969.

[465] G.I. Stassinopoulos. Optimal dynamic routing in multidestination networks. *IEEE Transactions on Communications*, Vol. COM-35, No. 4, pp. 472-475, April 1987.

[466] G.I. Stassinopoulos and P. Konstantopoulos. Optimal congestion control in single destination networks. *IEEE Transactions on Communications*, Vol. COM-33, No. 8, pp. 792-800, Aug. 1985.

[467] G.I. Stassinopoulos and H. Kukutos. Optimal dynamic routing in double ring networks. *IEEE Transactions on Communications*, Vol. 37, No. 8, pp. 890-896, Aug. 1989.

[468] G.I. Stassinopoulos and E.N. Protonotarios. Dynamic routing and prospects for on line implementation. *Proc. of International Communications Conference*, pp. 141-144, IEEE 1986.

[469] G.I. Stassinopoulos and E.N. Protonotarios. Optimal dynamic routing in multidestination networks. *IEEE Tr. on Communications*, Vol. COM-35, No. 4, pp. 472-475, Apr. 1987.

[470] T.E. Stern. A class of decentralized routing algorithms using relaxation. *Proc. National Telecommunications Conference*, pp. 42.1.1-42.1-5, IEEE 1976.

[471] B.W. Stuck and E. Arthurs. *A Computer and Communications Network Performance Analysis Primer.* Prentic-Hall, Inc., 1985.

[472] S. Sumita. Synthesis of an output buffer management scheme in a switching system for multimedia communications. *Proc. Infocom 90 Conference*, pp. 1226-1233, IEEE 1990.

[473] R.S. Sutton and A.G. Barto. Toward a modern theory of adaptive networks: expectation and prediction. *Psychological Review*, Vol. 88, No. 2, pp. 135-170, 1981.

[474] F. Szidarovsky and S. Yakowitz. A new proof of the existence and uniqueness of the Cournot equilibrium. *International Economic Review*, Vol. 18, No. 3, pp. 787-789, Oct. 1977.

[475] T. Szymanski. An analysis of "Hot-Potato" routing in a fiber optic packet switched hypercube. *Proc. Infocom 90 Conference*, pp. 918-925, IEEE 1990.

[476] D. Tabak and B.C. Kuo. *Optimal Control by Mathematical Programming.* Prentice Hall, 1971.

[477] L. Takacs. *Introduction to the Theory of Queues.* Oxford University Press, 1962.

[478] A.S. Tanebaum. *Computer Networks.* Prentice Hall, 2nd ed., 1988.

[479] A.N. Tantawi and D. Towsley. Optimal static load balancing in distributed computer systems. *Journal of A.C.M., Vol. 32, No 2*, pp. 445-465, April 1985.

[480] L. Tesfatsion. Pure strategy Nash equilibrium points and the Lefschetz fixed point theorem. *International Journal of Game Theory*, Vol. 12, No. 3, pp. 181-191, 1983.

[481] G.H. Thaker and J.B. Cain. Interactions between routing and flow control algorithms. *IEEE Transactions on Communications*, Vol. COM-34, No. 3, pp. 269-277, March 1986.

[482] M.A.L. Thathachar and P.S. Sastry. A new approach to the design of reinforcement schemes for learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 1, pp. 168-175, Jan./Febr. 1985.

[483] A. Thomasian. A performance study of dynamic load balancing in distributed systems. *Proc. Int. Conf. on Distributed Computing Systems*, pp. 178-184, IEEE 1987.

[484] H.C. Tijms. *Stochastic Modelling and Analysis: A Computational Approach.* J.Wiley & Sons, 1986.

[485] D. Tipper and M.K. Sundareshan. An optimal control approach to decentralized dynamic virtual circuit routing in computer networks. *Proc. IEEE Infocom 90*, pp. 926-933, IEEE 1990.

[486] M.J. Todd. *The Computation of Fixed Points and Applications.* Springer-Verlag, 1976.

[487] D. Towsley and R. Mirchandaney. The effect of communication delays on the performance of load balancing policies in distributed systems. *Computer Performance and Reliability*, G. Iazeolla et.al. (eds), pp. 213-226, Elsevier Science Publ. 1988.

[488] K.S. Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications.* Prentice-Hall, Inc., 1982.

[489] W.K. Tsai. Convergence of gradient projection routing methods in an asynchronous stochastic quasi-static virtual circuit network. *IEEE Tr. on Automatic Control*, Vol. AC-34, No. 1, pp. 20-33, Jan. 1989.

[490] W.K. Tsai, P.E. Cantrell, and J. Goos. Fairness of optimal routing in virtual circuit data networks. *Proc. of IEEE infocom 89*, pp. 119-126, 1989.

[491] W.K. Tsai, G. Huang, J.K. Antonio, and W.T. Tsai. Distributed iterative aggregation algorithms for box-constrained minimization problems and optimal routing in data networks. *IEEE Transactions on Automatic Control*, Vol. 34, No. 1, pp. 34-46, Jan. 1989.

[492] W.K. Tsai, J.N. Tsitsiklis, and D.P. Bertsekas. Some issues in distributed asynchronous routing in virtual circuit data networks. *Proc. IEEE 25th Conf. on Decision and Control*, pp.1335-1337, IEEE Dec. 1986.

[493] W.T. Tsai, C.V. Ramamoorthy, W.K. Tsai, and O. Nishiguchi. An adaptive hierarchical routing protocol. *Transactions on Computers*, Vol. 38, No. 8, pp. 1059-1074, Aug. 1989.

[494] M.L. Tsetlin. *Automaton Theory and Modeling of Biological Systems*. Academic Press, 1973.

[495] M.L. Tsetlin. On the behavior of finite automata in random media. *Avtomatika i Telemekhanika*, Vol. 22, No. 10, pp. 1345-1354, Oct. 1961.

[496] J.N. Tsitsiklis. Convexity and characterization of optimal policies in a dynamic routing problem. *J. of Optimization Theory and Applications*, Vol. 44, No. 1, pp. 105-136, Sept. 1984.

[497] J.N Tsitsiklis and D.P. Bertsekas. Distributed asynchronous optimal routing in data networks. *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 4, pp. 325-332, Apr. 1986.

[498] Y.Z. Tsypkin and A.S. Poznyak. Finite learning automata. *Engineering Cybernetics*, Vol. 10, pp. 478-490, May-June 1972.

[499] M. Tu and G.P. Papavassilopoulos. Performance versus informativeness in linear-quadratic Gaussian noncooperative games. *J. of Optimization Theory and Control*, Vol. 57, No. 1, pp. 161-187, April 1988.

[500] M. Tu and G.P. Papavassilopoulos. Impact of explicit and implicit control sharing on the performance of two-person one-act LQG Nash games. *Large Scale Systems*, Vol. 7, pp. 219-226, 1984.

[501] M. Tu and G.P. Papavassilopoulos. On the informational properties of the Nash solution of LQG dynamic games. *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 4, pp. 377-385, April 1985.

[502] P.P. Varaiya. *Notes on Optimization*. Van Nostrand Reinhold Co., 1972.

[503] V.I. Varshavskii and I.P. Vorontsova. On the behavior of stochastic automata with a variable structure. *Avtomatika i Telemekhanika*, Vol. 24, No. 3, pp. 353-360, March 1963.

[504] J. von Neuman and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 3rd ed., 1953.

[505] J. Walrand. *An Introduction to Queueing Networks*. Prentice Hall, 1988.

[506] Y.-T. Wang and R.T.J. Morris. Load sharing in distributed systems. *IEEE Trans. on Computers, Vol C-34, No 3*, pp. 204-217, March 1985.

[507] J.G. Wardrop. Some theoretical aspects of road traffic research. *Proc. Inst. Civil Engineers*, Part III, 1, pp. 325-378, 1952.

[508] R.R. Weber. On the optimal assignment of customers to parallel servers. *J. Applied Probability*, Vol. 15, pp. 406-413, 1978.

[509] L. Wei, Y. Yiouwei, and H. Zheng. A distributed routing algorithm in hybrid circuit/packet switching networks. *Proc. Infocom 88 Conference*, pp. 791-795, IEEE 1988.

[510] A. Weinrib and S. Shenker. Greed is not enough: Adaptive load sharing in large heterogeneous systems. *Proc. Infocom 88*, pp. 986-994, IEEE 1988.

[511] W. Whitt. Deciding which queue to join: some counterexamples. *Operations Research*, Vol. 34, No. 1, pp. 55-62, Jan.-Febr. 1986.

[512] J.B. Williams. Stability in noncooperative games. *Operations Research*, Vol. 19, pp. 774-783, 1971.

[513] R.J. Williams. Sufficient conditions for Nash equilibria in n-person games over reflexive Banach spaces. *J. of Optimization Theory and Applications*, Vol. 30, No. 3, pp. 383-394, March 1980.

[514] R. Wilson. Computing equilibria of n-person games. *SIAM J. Applied Mathematics*, Vol. 21, No. 1, pp. 80-87, July 1971.

[515] W. Winston. Optimality of the shortest line discipline. *J. Applied Probability*, Vol. 14, pp. 181-189, 1977.

[516] R.W. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall. 1989.

[517] G.M. Woodruff and R. Kositpaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE J. on Selected Areas in Communications*, Vol. 8, No. 3, pp. 437-446, Aprl 1990.

[518] E.F. Wunderlich. An analysis of dynamic virtual circuit routing. *Proc. IEEE National Telecommunications Conference*, pp. A3.3. - A3.3.6, IEEE 1981.

[519] E.F. Wunderlich, L. Kaufman, and B. Gopinath. The control of store and forward congestion in packet switched networks. *Proc. Int. Conference on Computer Communications*, pp. 851-856, 1980.

[520] P.S. Yu, S. Balsamo, and Y.-H. Lee. On dynamic load sharing and transaction routing. *Proc. of International Computer Symposium*, pp. 1265-1271, 1986.

[521] P.S. Yu, S. Balsamo, and Y.-H. Lee. Dynamic load sharing in distributed database systems. *Proc. Fall Joint Computer Conference*, pp. 675-683, IEEE 1986.

[522] P.S. Yu, S. Balsamo, and Y.-H. Lee. Dynamic transaction routing in distributed database systems. *IEEE transactions on Software Engineering*, Vol. 14, No. 9, pp. 1307-1318, Sept. IEEE 1988.

[523] T. Yum. A semidynamic deterministic routing rule in computer communication networks. *Proc. National Telecommunications Conference*, pp. 34.4.1-34.4.7, IEEE 1979.

[524] T. Yum and M. Schwartz. Packet-switched performance with different circuit-switched routing procedures in nonhierarchical integrated circuit-switched and packet-switched networks. *IEEE Transactions on Communications*, Vol. COm-35, No. 3, pp. 362-366, 1987.

[525] T.-S. Yum and M. Schwartz. Comparison of adaptive routing algorithms for computer communication networks. *Proc. National Telecommunications Conference*, pp. 4.1.1-4.1.4, IEEE 1978.

[526] T.-S. Yum and M. Schwartz. The join-biased-queue rule and its application to routing in computer communication networks. *IEEE Transactions on Communications*, Vol. COM-29, No. 4, pp. 505-511, Apr. 1981.

[527] T.-S.P. Yum. The design and analysis of a semidynamic deterministic routing rule. *IEEE Transactions on Communications*, Vol. COM-29, No. 4, pp. 498-504, Apr. 1981.

[528] T.P. Yum and H.-C. Lin. Adaptive load balancing for parallel queues. *International Communications Conference*, pp. 1268-1271, IEEE 1984.

[529] W.I. Zangwill. *Nonlinear Programming*. Prentice Hall, 1969.

[530] C. Zhou and B.S. Maglaris. Adaptive routing in computer networks within two-level procedures: delay performance and effect on network flows. *Proc. GLOBECOM*, pp. 746-752, IEEE 1985.

[531] S. Zhou. A trace-driven simulation study of dynamic load balancing. *IEEE Transactions on Software Engineering*, Vol. 14, No. 9, pp. 1327-1341, Sept. 1988.

[532] S. Zhou and D. Ferrari. A measurement study of load balancing performance. *Proc. Int. Conference on Distributed Computing Systems*, pp. 490-497, IEEE 1987.

[533] J. Zinky, G. Vichniac, and A. Khanna. Performance of the revised routing metric in the ARPANET and MILNET. *Proc. MILCOM*, pp. 219-224, IEEE 1989.