Load Sharing, Routing And Congestion Control in Distributed And Computing Systems As A Nash Game

A.A. Economides and John Silvester

CENG Technical Report 90-06

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213)740-4579

June 1990

DYNAMIC ROUTING AND CONGESTION CONTROL FOR MULTI-CLASS VIRTUAL CIRCUIT NETWORKS [1]

by

A. A. Economides, P. A. Ioannou and J. A. Silvester

Technical Report # 90-06-01, June 1990

Electrical Engineering - Systems Department
University of Southern California
University Park, Los Angeles, CA 90089-0781

[1] This work was supported by the National Science Foundation under Grant DMC-8452002 together with matching funds from AT & T Information Systems.

DYNAMIC ROUTING AND CONGESTION CONTROL FOR MULTI-CLASS VIRTUAL CIRCUIT NETWORKS

by

A. A. Economides, P. A. Ioannou and J. A. Silvester

Technical Report # 90-06-01, June 1990

Electrical Engineering - Systems Department
University of Southern California
University Park, Los Angeles, CA 90089-0781

Abstract

The decentralized adaptive joint routing and congestion control problem for multi-class multi-destination dynamic virtual circuit networks is considered.

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 is introduced. Then a multi-objective cost function of rejecting, setting up & maintaining virtual circuits, as well as of the packet delay and throughput is defined.

The combined problem is formulated 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, the state dependent routing and congestion controls are derived. 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 algorithm.

1 INTRODUCTION

Most existing networks (Codex, Euronet, SNA, Telenet, Transpac, Tymnet, etc.) as well as proposals for future high speed network architectures employ virtual circuit switching [3, 15, 21, 27, 44]. For each call (virtual circuit, or session, or transaction etc.), a single path is set up from source to destination and all entities (bursts, packets, cells, etc.) that belong to this call follow this path. Virtual circuit switching provides the following advantages:

1) Flexible resource management, since packets of each connection are on a specific path and not all over the network.

2) Easier and fairer access, service, accounting and billing control.

3) No packet resequencing at the destination (due to different delays of packets that arrive at the destination through different network paths), since packets belonging to a specific virtual circuit follow a single path from source to destination (hop-by-hop resequencing due to transmission errors may still needed).

4) Less packet header overhead, since the header carries only the virtual circuit number in which the packet belongs, and not the source and destination addresses.

5) No packet looping, since packets follow an already established path.

6) Less routing update overhead, since the routing is done on a per virtual circuit basis and not on a per packet basis.

7) Easier congestion and flow control for each connection by accepting a new virtual circuit only if it will not congest the network and by controlling the packet rate and resource usage of each admitted virtual circuit.

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 this paper, we consider the combined virtual circuit dynamic routing and congestion control problem. We give decentralized adaptive virtual circuit routing and congestion control algorithms that can be implemented in a probabilistic or deterministic fashion. In the next section, we describe the different approaches that have been taken for the virtual circuit routing problem in previous studies.

2 PREVIOUS STUDIES

Depending on the assumptions of how fast the network dynamics change, three main approaches have been investigated for virtual circuit routing:

1) In *Static* virtual circuit routing, the number of virtual circuits is given (i.e. no virtual circuit arrivals) and the packet arrival rate per virtual circuit is also given. There exist two main formulations for the static virtual circuit routing problem:

1.1) In the *continuous nonlinear programming* formulation, the decision variables are the flows on the links.

Gerla & Nilsson [23], Gerla,Chan & Boisson De Marca [22], Lam & Lien [35, 36], Kobayashi & Gerla [34], De Souza & Gerla [8] model a virtual circuit network as a closed queueing network where each closed chain corresponds to a flow controlled virtual circuit. Packet arrivals belonging to a specific virtual circuit that find its virtual circuit window full are lost. They use *flow deviation* - type algorithms and report numerical results.

1.2) In the *0-1 nonlinear programming* formulation the decision variables are the assignment or not of a given virtual circuit to a path.

Courtois & Semal [7] modify the *flow deviation* method for the non bifurcated flow case. Gavish & Hantler [19], Gavish & Neuman [20], Narasimhan, Pirkul & De [41], Lin & Yee [38], use *Lagrangean relaxation & subgradient optimization* techniques and report numerical results.

2) In *Quasi-static* virtual circuit routing, it is assumed that the externally arriving traffic changes very slowly over time and individual offered traffic sample functions do not exhibit frequently large and persistent deviations from their averages [4, 51, 3]. Therefore the virtual circuit routing decisions may depend only on the input and link flows.

For the case of "many small users" [17, 18, 51, 3], where there is a very large number of virtual circuits between each source-destination pair and each virtual circuit has a very small packet arrival rate, Gafni & Bertsekas [18] show that the routing updating period should be much smaller than the average virtual circuit duration. In this case, Bertsekas [4], Tsitsiklis & Bertsekas [51], Tsai, Tsitsiklis & Bertsekas [50], Tsai [49, 48] formulate the virtual circuit routing problem as a *continuous nonlinear programming* problem with decision variables the flows on the links or the paths. They use *gradient projection* - type algorithms to precompute the optimum flows. Then they assign each new arriving virtual circuit on the paths to achieve these optimum flows. Tsai [49] shows that deterministic assignment of the new arriving virtual circuits is better than random.

Humblet, Soloway & Sleinka [28] define as link length in the Codex network [28, 3], the difference of the link cost function when a new virtual circuit is routed through this link and when not. Then a new virtual circuit is routed along the minimum length path. Gopal, Kadaba & Wieber [25] define as link length the first derivative of the average number on this link with respect to the flow through this link. The link length is updated only when the link utilization crosses a threshold. Then a new virtual circuit is also routed along the minimum length path. Jaffe & Segall [29] minimize the average delay over a range of arrival rates with respect to link utilization thresholds and link lengths. Gersht [24] assumes that the average virtual circuit duration is much larger than the average packet delay in the network for the Telenet network. At every node, a new arriving virtual circuit is routed through the outgoing link with the minimum number of virtual circuits.

3) In *Adaptive* virtual circuit routing it is assumed that the network state is continuously changing due to real time traffic fluctuations. Therefore, the routing decisions are taken for each virtual circuit independently and depend on the current network state, for example the current network topology, the current number of packets & virtual circuits, the current virtual circuit & packet arrival rates, the current service requirements, the current link error rates etc.

Such a stochastic problem is extremely difficult even under Markovian assumptions. The resulting Markov Chain does not have product form solution because the transition probabilities depend on the network state. Also, since an adaptive routing algorithm should depend on the current network state we must find the transient solution of the corresponding Markov Chain with time dependent external arrival and service rates (recall the nasty expressions for the transient analysis of a simple $M/M/1$ queue [33]). Finally, even if we solve the corresponding Markov Chain and find state dependent controls that depend on the current network state, it is impossible to know the current network state at all network resources. The needed time for the network state observations to be transferred from one network point to another is a random variable that also depends on the network state and during this time the network state has already changed. Furthermore, it is also difficult to obtain accurate estimates of the instantaneous rates. So, the network state information is always obsolete and inaccurate. Therefore attacking the stochastic problem directly would be difficult,

Also, in real network control implementations, the average rather than the instantaneous measures of the network state are used due to the following reasons: 1) wide variability of the instantaneous network state values, 2) obsolete network state information, due to transfer delay, 3) periodic implementation of the network control, 4) communication overhead in transferring the instantaneous network state information, and 5) computation overhead in calculation for an exact network optimization.

Two main formulations exist for the adaptive virtual circuit routing problem:

3.1) In the *stochastic learning automata* formulation, the fact that our information about the network state is inaccurate is incorporated into the routing decisions.

Economides, Ioannou & Silvester [11, 10] introduce *state-dependent multiple reward / penalty* stochastic learning automata updating schemes and use them for virtual circuit routing. The idea here is not to overreact by completly trusting the information we have about the network state. Instead of using a definitive decision as to where to send a new arriving virtual circuit, we vary the routing probabilities strongly favoring the minimum length path, i.e. we have a "probabilistic selection of the minimum length path". Note that, the routing along the minimum length path (with probability 1) is a special case of the stochastic learning automata routing. They define as link length the unfinished work due to packets and virtual circuits currently on this link (for user optimization) or the increase in the current number of packets due to the addition of a new virtual circuit on this link (for system optimization).

3.2) In the *optimal control* formulation, the network state dynamic evolution is described by a state space model of difference or differential equations.

Gafni & Bertsekas [17] consider a simple stochastic problem that turns out to be an NP-complete problem. Then they transform it to the corresponding deterministic problem that is still a hard problem. Finally, they suggest a discrete time heuristic - routing along the minimum first derivative length path.

Tipper & Sundareshan [46, 47] and Economides, Ioannou & Silvester [9] take a different approach. They describe the network state using nonlinear deterministic dynamic models that are based on queueing theory concepts and they use Pontryagin's maximum principle to provide the necessary optimality conditions.

Tipper & Sundareshan [46, 47] consider a single source-destination virtual circuit routing problem. A virtual circuit is to be routed along one of the paths between its source and its destination. They develop a nonlinear dynamic queueing model for the average number of packets of this virtual circuit and of interfering traffic at each link along the paths from its source to its destination. Then they define as cost function the average number of packets of this virtual circuit and of the interfering traffic on every link along the paths from its source to its destination. Pontryagin's maximum principle provides the necessary conditions for optimal virtual circuit routing. They propose routing of this virtual circuit along the minimum length path, where the length of a link is a quadratic function of the average number of packets on it.

Economides, Ioannou & Silvester [9] consider the routing and congestion control problem for window flow controlled single source-destination virtual circuit networks with error prone links. They develop a queueing model for each error controlled and window flow controlled link. They express the link capacity as a function of the link error rate, link transmission rate, link propagation delay, ACK delay, time-out delay and other flow control parameters. They introduce dynamic nonlinear queueing models that describe the dynamic interaction of the virtual circuit and packet processes. Then they define a multi-objective function of the average number of virtual circuits and packets on every link, the virtual circuit rejection flow and the packet throughput on every link. Pontryagin's maximum principle provides the necessary optimality conditions that are also sufficient for this convex virtual circuit routing and congestion control problem. A new virtual circuit is admitted into the network if the cost that it will produce is less then the profit that it will offer. Then it is routed along the minimum length path. The length of a link is a quadratic function of the average number of packets on it and a linear function of the average number of packets per virtual circuit, of the cost per virtual circuit and of the packet throughput profit.

In this paper, we consider the combined routing and congestion control problem in multi-destination multi-class dynamic virtual circuit networks. The network dynamics occur at two different time scales. The virtual circuit process evolves at the slower time scale and is used in the evolution of the packet process that occurs at the faster time scale. We introduce a nonlinear dynamic queueing model for the average number of different class virtual circuits on every link, and a nonlinear dynamic queueing model for the average number of different class packets on every link. These models are coupled together and describe the dynamic interaction among the multi-destination multi-class virtual circuit and packet processes. We set up a multi-objective cost function of rejecting, setting up and maintaining virtual circuits, as well as of the packet delay and throughput. Then we formulate the combined routing and congestion control problem for dynamic multi-destination multi-class virtual circuit networks as an optimal control problem. Pontryagin's maximum principle provides necessary optimality conditions that are also sufficient. For the finite horizon problem, a Two-Point Boundary-Value Problem must be solved numerically. For the long-run stationary equilibrium, we derive state dependent routing and congestion controls.

Finally, if an algorithm is to be used for on-line routing in a real network, it should be at least tested in a simulation program. Therefore, we compare a simple case of the proposed routing algorithm with a shortest queue routing and the optimal quasi-static routing algorithm via simulation and we discuss the parameters that affect the routing decisions.

3 VIRTUAL CIRCUIT NETWORK MODEL

Consider an arbitrary network topology with multiple classes of virtual circuit traffic between multiple source-destination pairs (Fig. 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 inroduce 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 (Fig. 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 $\gamma_{[sd]}(t) \geq 0$ (Fig. 3). 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 $1/\delta_{[sd]}(t)$ (Fig. 4). We note that thousands of virtual circuits can co-exist on a link (well within today's technology capabilities) [31].

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, $\delta_{[sd]}(t) * V_{ij[sd]}(t) * \Delta t$ (Fig. 5). So, the

$[sd]$ virtual circuit process at link ij is described by

$$V_{ij[sd]}(t+\Delta t) = V_{ij[sd]}(t) + \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) * \Delta t - \delta_{[sd]}(t) * V_{ij[sd]}(t) * \Delta t \qquad \forall ij \quad \forall[sd]$$

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

$$\dot{V}_{ij[sd]}(t) = \lim_{\Delta t\to 0} \frac{V_{ij[sd]}(t+\Delta t) - V_{ij[sd]}(t)}{\Delta t} \qquad \forall ij \quad \forall[sd]$$

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) (Fig. 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 [9]. Packets are serviced according to first-come-first-served or processor sharing scheduling. Katevenis [31] and Morgan [40] 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 $M/M/1$ (Fig. 2) or as a Processor Sharing queue (Fig. 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 $\mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t))$.

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

$$N_{ij[sd]}(t+\Delta t) = N_{ij[sd]}(t) + r_{[sd](t)} * V_{ij[sd]}(t) * \Delta t - \mu_{ij}(t) * \rho_{ij[sd]}(t)(\mathbf{N}_{ij}(t)) * \Delta t \qquad \forall ij \quad \forall [sd]$$

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

$$f_{V,ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij \in \pi[sd]}(t) - \delta_{[sd]}(t) * V_{ij[sd]}(t) \quad \forall ij \quad \forall [sd]$$

$$f_{N,ij[sd]}(t) = r_{[sd]}(t) * V_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}_{ij}(t)) \quad \forall ij \quad \forall [sd]$$

$$\mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t)) = \begin{bmatrix} \cdots \\ f_{V,ij[sd]}(t) \\ f_{N,ij[sd]}(t) \\ \cdots \end{bmatrix}$$

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 6.

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.

4 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. An optimal control approach has also been taken by Filipiak [12, 13, 14] for the single-class single-destination datagram routing problem, Tipper & Sundareshan [46, 47] and Economides, Ioannou & Silvester [9] for the single-class single-destination virtual circuit routing 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 maximimize 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

$$\mathbf{V} = \{ \quad \phi_{o[sd]}(t),\ \phi_{\pi[sd]}(t)\ \ \forall\pi[sd]\ \ \forall[sd], \qquad \textit{such that}$$

$$\phi_{o[sd]}(t) \geq 0,\ \ \phi_{\pi[sd]}(t) \geq 0\ \ \forall\pi[sd]\ \ \forall[sd], \qquad \phi_{o[sd]}(t) + \sum_{\pi[sd]} \phi_{\pi[sd]}(t) = 1\ \ \forall[sd]\ \}$$

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 Adaptive Virtual Circuit Routing and Congestion Control problem *(AVCRCC)* is:

Problem AVCRCC :

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

$$\textit{with respect to} \qquad \mathbf{U}(t)$$

$$\textit{such that} \qquad \dot{\mathbf{X}}(t) = \mathbf{f}(t, \mathbf{X}(t), \mathbf{U}(t))$$

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

$$\mathbf{X}(t_f)\ \textit{free}$$

$$\mathbf{U}(t) \in \mathbf{V}$$

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 [6, 2, 5, 16, 52].

Theorem 1. Necessary conditions

Let $\mathbf{U}^*(t)$ be a piecewise continous control defined on $[t_0, t_f]$ which solves Problem AVCRCC 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 $t \in [t_0, t_f]$,

$$\phi^*_{o[sd]}(t) \begin{cases} > 0 & \textit{only if } C_{o[sd]}(t) = \min\{C_{o[sd]}, \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & o.w. \ \forall [sd] \end{cases}$$

$$\phi^*_{\pi[sd]}(t) \begin{cases} > 0 & \textit{only if } \sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in \pi[sd]}(t) = \min\{C_{o[sd]}(t), \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & o.w. \ \forall \pi[sd] \ \ \forall [sd] \end{cases}$$

$$\dot{V}^*_{ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi^*_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) - \delta_{[sd]}(t) * V^*_{ij[sd]}(t) \quad \forall ij \quad \forall [sd]$$

$$\dot{N}^*_{ij[sd]}(t) = r_{[sd]}(t) * V^*_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}^*_{ij}(t)) \quad \forall ij \quad \forall [sd]$$

$$V^*_{ij[sd]}(t_0) = V_{ij[sd],0} \quad \forall ij \quad \forall [sd]$$
$$N^*_{ij[sd]}(t_0) = N_{ij[sd],0} \quad \forall ij \quad \forall [sd]$$

$$\dot{Q}_{V,ij[sd]}(t) = -\{\, C_{V,ij[sd]}(t) - P_{V,ij[sd]}(t) * \delta_{[sd]}(t) + P_{N,ij[sd]}(t) * r_{[sd]}(t) \,\} \quad \forall ij \quad \forall [sd]$$

$$\dot{Q}_{N,ij[sd]}(t) = -\{\, C_{N,ij[sd]}(t) - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)} -$$
$$- \sum_{[s_1 d_1]} P_{N,ij[s_1 d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)} \,\} \quad \forall ij \quad \forall [sd]$$

$$P_{V,ij[sd]}(t_f) = 0 \quad \forall ij \quad \forall [sd]$$
$$P_{N,ij[sd]}(t_f) = 0 \quad \forall ij \quad \forall [sd]$$

Proof:

The Hamiltonian must satisfy the following condition

$H(t, \mathbf{X}^*(t), \mathbf{U}^*(t), \mathbf{P}(t)) \le H(t, \mathbf{X}^*(t), \mathbf{U}, \mathbf{P}(t)) \quad \forall\ \mathbf{U} \in \mathbf{V}$

which is equivalent to the following condition

$$\sum_{[sd]} \left\{ \gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi^*_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi^*_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t)] \right\} \le$$
$$\le \sum_{[sd]} \left\{ \gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t)] \right\}$$

$\forall\ \phi_{o[sd]},\ \phi_{\pi[sd]} \in \mathbf{V} \quad \forall \pi[sd] \quad \forall [sd]$

Since, there is no dependency among the controls for different source-destination pairs $[sd]$, we can decomposed the above conditions $\forall [sd]$ to

$$\gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi^*_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi^*_{ij[sd]}(t) * 1_{ij\in\pi[sd]}(t)] \le$$

$$\leq \gamma_{[sd]}(t) * [C_{o[sd]}(t) * \phi_{o[sd]}(t) + \sum_{\pi[sd]} \sum_{ij} P_{V,ij[sd]}(t) * \phi_{ij[sd]}(t) * 1_{ij \in \pi[sd]}(t)]$$

$\forall\ \phi_{o[sd]},\ \phi_{\pi[sd]} \in \mathbf{V} \quad \forall \pi[sd]$

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

$$\begin{aligned} \dot{V}^*_{ij[sd]}(t) &= \sum_{\pi[sd]} \gamma_{[sd]}(t) * \phi^*_{\pi[sd]}(t) * 1_{ij \in \pi[sd]}(t) - \delta_{[sd]}(t) * V^*_{ij[sd]}(t) \quad \forall ij\ \ \forall [sd] \\ \dot{N}^*_{ij[sd]}(t) &= r_{[sd]}(t) * V^*_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\mathbf{N}^*_{ij}(t)) \quad \forall ij\ \ \forall [sd] \end{aligned}$$

The optimal state must also satisfy the initial state $\mathbf{X}^*(t_0) = \mathbf{X}_0$, therefore

$$\begin{aligned} V^*_{ij[sd]}(t_0) &= V_{ij[sd],0} \quad \forall ij\ \ \forall [sd] \\ N^*_{ij[sd]}(t_0) &= N_{ij[sd],0} \quad \forall ij\ \ \forall [sd] \end{aligned}$$

The costate variables must satisfy the following conditions

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

which can be rewritten as

$$\dot{Q}_{V,ij[sd]}(t) = -\frac{\partial H(t, \mathbf{X}^*(t), \mathbf{U}^*(t), \mathbf{P}(t))}{\partial V_{ij[sd]}(t)} =$$

$$= -\{\ C_{V,ij[sd]}(t) - P_{V,ij[sd]}(t) * \delta_{[sd]}(t) + P_{N,ij[sd]}(t) * r_{[sd]}(t)\ \} \quad \forall ij \quad \forall [sd]$$

$$\dot{Q}_{N,ij[sd]}(t) = -\frac{\partial H(t, \mathbf{X}^*(t), \mathbf{U}^*(t), \mathbf{P}(t))}{\partial N_{ij[sd]}(t)} =$$

$$= -\{\ C_{N,ij[sd]}(t) - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)} -$$

$$- \sum_{[s_1 d_1]} P_{N,ij[s_1 d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij}(t))}{dN_{ij[sd]}(t)}\ \} \quad \forall ij \quad \forall [sd]$$

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 [39, 30, 45, 42].

Theorem 2. *Sufficient conditions*

Let $(\bar{\mathbf{X}}(t), \bar{\mathbf{U}}(t))$ be an admissible pair in Problem AVCRCC. 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 $\mathbf{P}(t) = [...P_{V,ij[sd]}(t)\ P_{N,ij[sd]}(t)...]^T$ such that the following conditions are satisfied for all $t \in [t_0, t_f]$

$$\bar{\phi}_{o[sd]}(t) \begin{cases} > 0 & \text{only if } C_{o[sd]}(t) = \min\{C_{o[sd]}(t), \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & o.w. \ \forall [sd] \end{cases}$$

$$\bar{\phi}_{\pi[sd]}(t) \begin{cases} > 0 & \text{only if } \sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in \pi[sd]}(t) = \min\{C_{o[sd]}(t), \min_{p[sd]}\{\sum_{ij} P_{V,ij[sd]}(t) * 1_{ij \in p[sd]}(t)\}\} \\ = 0 & o.w. \ \forall \pi[sd] \ \ \forall [sd] \end{cases}$$

$$\dot{\bar{V}}_{ij[sd]}(t) = \sum_{\pi[sd]} \gamma_{[sd]}(t) * \bar{\phi}_{\pi[sd]}(t) * 1_{ij\in\pi[sd]}(t) - \delta_{[sd]}(t) * \bar{V}_{ij[sd]}(t) \quad \forall ij \quad \forall[sd]$$

$$\dot{\bar{N}}_{ij[sd]}(t) = r_{[sd]}(t) * \bar{V}_{ij[sd]}(t) - \mu_{ij}(t) * \rho_{ij[sd]}(\bar{\mathbf{N}}_{ij}(t)) \quad \forall ij \quad \forall[sd]$$

$$\bar{V}_{ij[sd]}(t_0) = V_{ij[sd],0} \quad \forall ij \quad \forall[sd]$$
$$\bar{N}_{ij[sd]}(t_0) = N_{ij[sd],0} \quad \forall ij \quad \forall[sd]$$

$$\dot{Q}_{V,ij[sd]}(t) = -\{ \ C_{V,ij[sd]}(t) - P_{V,ij[sd]}(t) * \delta_{[sd]}(t) + P_{N,ij[sd]}(t) * r_{[sd]}(t) \ \} \quad \forall ij \quad \forall[sd]$$

$$\dot{Q}_{N,ij[sd]}(t) = -\{ \ C_{N,ij[sd]}(t) - \sum_{[s_1d_1]} C_{\mu,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\bar{\mathbf{N}}_{ij}(t))}{dN_{ij[sd]}(t)} - $$
$$- \sum_{[s_1d_1]} P_{N,ij[s_1d_1]}(t) * \mu_{ij}(t) * \frac{d\rho_{ij[s_1d_1]}(\bar{\mathbf{N}}_{ij}(t))}{dN_{ij[sd]}(t)} \ \} \quad \forall ij \quad \forall[sd]$$

$$P_{N,ij[sd]}(t) \geq 0 \quad \forall ij \quad \forall[sd]$$

$$P_{V,ij[sd]}(t_f) = 0 \quad \forall ij \quad \forall[sd]$$
$$P_{N,ij[sd]}(t_f) = 0 \quad \forall ij \quad \forall[sd]$$

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 $\mathbf{X}(t)$ [39, 30, 45, 42]).

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, $\rho_{ij[sd]}(\mathbf{N}_{ij}) =$

$\frac{N_{ij[sd]}}{1+\sum\limits_{[s_1d_1]} \rho_{ij[s_1d_1]}}$ is defined for $\mathbf{N}_{ij} \geq 0$, is concave, monotonically increasing and twice differentiable in $\mathbf{N}_{ij}$ with $\lim\limits_{\mathbf{N}_{ij} \to \infty} \rho_{ij}(\mathbf{N}_{ij}) = 1$.

So, after numerically solving a two-Point Boundary-Value Problem (TPBVP), we have the optimal congestion control and routing decisions. Numerical methods [1, 6, 26, 32, 37, 43] 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 paper, 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 STATE DEPENDENT ROUTING & CONGESTION CONTROLS

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

$$
\begin{aligned}
\textit{minimize} \quad & \sum_{[sd]} C_{o[sd]} * \gamma_{[sd]} * \phi_{o[sd]} + \\
+ & \sum_{[sd]} \sum_{ij} C_{V,ij[sd]} * V_{ij[sd]} + \\
+ & \sum_{[sd]} \sum_{ij} C_{N,ij[sd]} * N_{ij[sd]} - \\
- & \sum_{[sd]} \sum_{ij} C_{\mu,ij[sd]} * \mu_{ij} * \rho_{ij[sd]}(\mathbf{N}_{ij})
\end{aligned}
$$

$$
\begin{aligned}
\textit{with respect to} \quad & \text{the congestion controls } \phi_{o[sd]} \geq 0 && \forall [sd] \\
& \text{the routing fractions } \phi_{\pi[sd]} \geq 0 && \forall \pi[sd] \;\; \forall [sd] \\
\textit{such that} \quad & 0 = \sum_{\pi[sd]} \gamma_{[sd]} * \phi_{\pi[sd]} * 1_{ij \in \pi[sd]} - \delta_{[sd]} * V_{ij[sd]} && \forall ij \;\; \forall [sd] \\
& 0 = r_{[sd]} * V_{ij[sd]} - \mu_{ij} * \rho_{ij[sd]}(\mathbf{N}_{ij}) && \forall ij \;\; \forall [sd] \\
& \phi_{o[sd]}, \;\; \phi_{\pi[sd]} \geq 0 && \forall \pi[sd] \;\; \forall [sd] \\
& \phi_{o[sd]} + \sum_{\pi[sd]} \phi_{\pi[sd]} = 1 && \forall [sd]
\end{aligned}
$$

The minimization of the Hamiltonian with respect to the congestion control and routing fractions is equivalent to the following minimization problem

$$\begin{array}{ll} \textit{minimize} & \sum_{[sd]} \left\{ \gamma_{[sd]} * [C_{o[sd]} * \phi_{o[sd]} + \sum_{\pi[sd]} \sum_{ij} Q_{V,ij[sd]} * \phi_{\pi[sd]} * 1_{ij \in \pi[sd]}] \right\} \\ \textit{with respect to} & \phi_{o[sd]}, \phi_{\pi[sd]}, \quad \forall \pi[sd] \quad \forall [sd] \\ \textit{such that} & \phi_{o[sd]} + \sum_{\pi[sd]} \phi_{\pi[sd]} = 1 \quad \phi_{o[sd]},\ \phi_{\pi[sd]} \geq 0 \quad \forall \pi[sd] \quad \forall [sd] \end{array}$$

where the costate variables $Q_{V,ij[sd]}$ for the expected number of virtual circuits and the costate variables $Q_{N,ij[sd]}$ 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

$$\begin{array}{ll} \textit{minimize} & \gamma_{[sd]} * [C_{o[sd]} * \phi_{o[sd]} + \sum_{\pi[sd]} \sum_{ij} Q_{V,ij[sd]} * \phi_{ij[sd]} * 1_{ij \in \pi[sd]}] \\ \textit{with respect to} & \phi_{o[sd]}, \phi_{\pi[sd]} \quad \forall \pi[sd] \\ \textit{such that} & \phi_{o[sd]} + \sum_{\pi[sd]} \phi_{\pi[sd]} = 1, \quad \phi_{o[sd]},\ \phi_{\pi[sd]} \geq 0 \quad \forall \pi[sd] \end{array}$$

Define the minimum cost at source node s for the $[sd]$ virtual circuit traffic to be $Q^*_{V,s[sd]} = \min\{C_{o[sd]}, \min_{p[sd]}\{\sum_{ij} Q_{V,ij[sd]} * 1_{ij \in p[sd]}\}\}$. Then the optimum congestion controls are:

$$\phi^*_{o[sd]} \begin{cases} > 0 & \text{only if } C_{o[sd]} = Q^*_{V,s[sd]} \\ = 0 & \text{o.w.} \end{cases}$$

and the optimum routing fractions are:

$$\phi^*_{\pi[sd]} \begin{cases} > 0 & \text{only if } \sum_{ij} Q_{V,ij[sd]} * 1_{ij \in \pi[sd]} = Q^*_{V,s[sd]} \\ = 0 & \text{o.w.} \end{cases}$$

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. $C_{o[sd]} = Q^*_{V,s[sd]}$. Also, path $\pi[sd]$ will be used for the $[sd]$ traffic only if its costate variable achieves the minimum cost, i.e. $\sum_{ij} Q_{V,ij[sd]} * 1_{ij \in \pi[sd]} = Q^*_{V,s[sd]}$.

When the congestion control and routing fractions achieve their optimum values, we have

$$C_{o[sd]} * \phi^*_{o[sd]} + \sum_{ij} \sum_{\pi[sd]} Q_{V,ij[sd]} * \phi^*_{ij[sd]} * 1_{ij\in\pi[sd]} = Q^*_{V,s[sd]}$$

The optimum congestion control and routing decisions depend on the values of the costate variables $Q_{V,ij[sd]}$ $\forall ij$ $\forall[sd]$. So, we have to calculate the costate variables $Q_{V,ij[sd]}$ for each link ij for each $[sd]$ traffic.

At steady state, the costate variables must satisfy $\dot{Q}_{V,ij[sd]} = 0 \quad \forall ij \quad \forall[sd]$.

$$\dot{Q}_{V,ij[sd]} = 0 \Rightarrow \quad -\{\, C_{V,ij[sd]} - Q_{V,ij[sd]} * \delta_{[sd]} + Q_{N,ij[sd]} * r_{[sd]} \,\} = 0 \quad \forall ij \quad \forall[sd]$$

Then

$$Q_{V,ij[sd]} \quad = \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * Q_{N,ij[sd]} \quad \forall ij \quad \forall[sd]$$

Next, in order to calculate the costate variables $Q_{V,ij[sd]}$ for the expected number of $[sd]$ virtual circuits, we must first calculate the costate variables $Q_{N,ij[sd]}$ for the expected number of $[sd]$ packets. At steady state, the costate variables must satisfy $\dot{Q}_{N,ij[sd]} = 0 \quad \forall ij \quad \forall[sd]$.

$$\dot{Q}_{N,ij[sd]} = 0 \Rightarrow \quad -\{\, C_{N,ij[sd]} - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}} -$$

$$- \sum_{[s_1 d_1]} Q_{N,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}} \,\} = 0 \quad \forall ij \quad \forall[sd]$$

In order to find the costate variables $Q_{N,ij[sd]}$ 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:

$$C_{N,ij[sd]} - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}} - \sum_{[s_1 d_1]} Q_{N,ij[s_1 d_1]} * \mu_{ij} * \frac{d\rho_{ij[s_1 d_1]}(\mathbf{N}^*_{ij})}{dN_{ij[sd]}} = 0 \quad \forall[sd]$$

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 $Q_{N,ij[sd]}$ costate variable system of equations for each link ij, as

$$C_{N,ij[sd]} - \sum_{[s_1 d_1]} C_{\mu,ij[s_1 d_1]} * \mu_{ij} * \left\{ \frac{1 + \sum_{[s_2 d_2] \neq [sd]} N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} - \sum_{[s_2 d_2] \neq [sd]} \frac{N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} \right\} -$$

$$- \sum_{[s_1 d_1]} Q_{N,ij[s_1 d_1]} * \mu_{ij} * \left\{ \frac{1 + \sum_{[s_2 d_2] \neq [sd]} N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} - \sum_{[s_2 d_2] \neq [sd]} \frac{N^*_{ij[s_2 d_2]}}{(1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2} \right\} = 0 \quad \forall [sd]$$

The solution to the above system is

$$Q_{N,ij[sd]} = \frac{1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]}}{\mu_{ij}} * \{ \quad C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) +$$

$$+ \sum_{[s_2 d_2] \neq [sd]} C_{N,ij[s_2 d_2]} * N^*_{ij[s_2 d_2]} \} - C_{\mu,ij[sd]} \quad \forall [sd]$$

In section 3, we stated that it must hold $Q_{N,ij[sd]} \geq 0$. 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, $C_{N,ij[s_2 d_2]} = C_{N,ij} \ \ \forall [s_2 d_2]$, the above solution simplifies to

$$Q_{N,ij[sd]} = \frac{C_{N,ij} * (1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]})^2}{\mu_{ij}} - C_{\mu,ij[sd]} \quad \forall [sd]$$

Substituting the $Q_{N,ij[sd]}$ into $Q_{V,ij[sd]}$, we have the cost to go from node s to destination d through path $\pi[sd]$:

$$Q_{V,\pi[sd]} = \sum_{ij\in\pi[sd]} \{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \{ \frac{1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]}}{\mu_{ij}} * \{ C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) + $$

$$+ \sum_{[s_2 d_2]\neq[sd]} C_{N,ij[s_2 d_2]} * N^*_{ij[s_2 d_2]} \} - C_{\mu,ij[sd]} \} \quad \forall\pi[sd] \quad \forall[sd]$$

The following Theorems follow immediately:

Theorem 3. Congestion Control

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*

$$C_{o[sd]} < \min_{p[sd]} \{ \sum_{ij\in p[sd]} \{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \{ \frac{1 + \sum_{[s_1 d_1]} N^*_{ij[s_1 d_1]}}{\mu_{ij}} * \{ C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) + $$

$$+ \sum_{[s_2 d_2]\neq[sd]} C_{N,ij[s_2 d_2]} * N^*_{ij[s_2 d_2]} \} - C_{\mu,ij[sd]} \} \} \}$$

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*

$$\sum_{ij\in\pi[sd]} \{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \{ \frac{1 + \sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} \{ C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) +$$

$$+ \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \} - C_{\mu,ij[sd]} \} =$$

$$= \min_{p[sd]} \{ C_{o[sd]}, \sum_{ij\in p[sd]} \{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \{ \frac{1 + \sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} * \{ C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) +$$

$$+ \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \} - C_{\mu,ij[sd]} \} \} \}$$

From the above analysis, we have derived that the length of each link ij for an $[sd]$ virtual circuit is

$$l_{ij[sd]} = \{ \frac{C_{V,ij[sd]}}{\delta_{[sd]}} + \frac{r_{[sd]}}{\delta_{[sd]}} * \{ \frac{1 + \sum_{[s_1d_1]} N^*_{ij[s_1d_1]}}{\mu_{ij}} \{ C_{N,ij[sd]} * (1 + N^*_{ij[sd]}) +$$

$$+ \sum_{[s_2d_2]\neq[sd]} C_{N,ij[s_2d_2]} * N^*_{ij[s_2d_2]} \} - C_{\mu,ij[sd]} \}$$

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.

6 DECENTRALIZED ADAPTIVE CONGESTION CONTROL AND ROUTING

In this section, we show how the optimum congestion controls and routing decisions can be implemented in an adaptive and decentralized fashion.

For adaptive virtual circuit network control both the congestion control and the routing decisions are done *for each virtual circuit* independently. In the previous section, we derived state dependent routing and congestion controls. As we mentioned in section 3, there are two general ways of achieving these congestion controls:

1) In *probabilistic congestion control*, each virtual circuit is accepted in a network node with very high probability if the cost of admitting it is less than the cost of rejecting it, otherwise it is rejected with high probability. One possible way to update the congestion control probabilities is using stochastic learning automata [11, 10].

2) In *deterministic congestion control*, each virtual circuit is accepted in a network node according to a deterministic rule. One possible rule is to achieve the congestion control fractions in a round-robin fashion. Another possible rule is to admit this virtual circuit if the cost of admitting it plus a threshold is less than the cost of rejecting it.

Similarly, there are two general ways of achieving the routing fractions:

1) In *probabilistic routing*, each virtual circuit is routed along the minimum length path with very high probability. One possible way to update the routing probabilities is using stochastic learning automata [11, 10].

2) In *deterministic routing*, each virtal circuit is routed along the minimum length path according to a deterministic rule. One possible rule is to achieve the routing fractions in a round-robin fashion. Another possible rule is to route the virtual circuit along the path for which its length plus a threshold is minimum.

We also want the congestion control and routing decisions to be done in a decentralized fashion. For the congestion controls, there are two decentralized implementation options:

1) In *source congestion control*, each source node s admitts or rejects an externally arriving $[sd]$ virtual circuit. So, each source node s evaluates the length of its paths $1[sd], ..., \pi[sd], ...$ to the destination d. If there is at least one length less than the cost of not admitting this $[sd]$ virtual circuit, then node s accepts this $[sd]$ virtual circuit, otherwise the source node s rejects it.

2) In *node-by-node congestion control*, each node i may reject an $[sd]$ virtual circuit passing through this node i. So, let paths $1[sd],...,\pi[sd], ...$ from source s to destination d pass through node i. Node i evaluates the length of these paths from i to d. If there is at least one length less than the cost of rejecting this $[sd]$ virtual circuit, then node i accepts this $[sd]$ virtual circuit, otherwise node i rejects it.

Although node-by-node congestion control provides complete decentralization, it may also waste a lot of network resources. If a virtual circuit is rejected at an intermediate node, then all the network resources used by this virtual circuit are wasted.

Similarly, there are two decentralized options for the virtual circuit routing decisions:

1) In *source routing*, each source node s selects the path to the destination d that will be followed by an $[sd]$ virtual circuit originating and admitted at this node s. So, each source node s evaluates the length of its paths $1[sd], ..., \pi[sd], ...$ to the destination d and routes the $[sd]$ virtual circuit along the minimum length paths.

2) In *node-by-node routing* (or *link-by-link routing*), each node i selects the next node for an $[sd]$ virtual circuit passing through this node i. So, each node i evaluates the length of the paths $1[sd],...,\pi[sd], ...$ from i to d and routes the $[sd]$ virtual circuit passing through i along the minimum length paths.

Although node-by-node routing provides complete decentralization, it also requires more overhead, since we need to keep the information about all paths between all sources to all destinations. In source routing, we only need to keep information about all paths from this source node to all destinations. An important advantage of source routing is that the source defines the paths that its traffic may follow avoiding (for example, for security reasons) some other nodes. In node-by-node routing the source node has no control over the path that its traffic may follow. Also, in source routing it is trivial to guarantee no looping. In node-by-node routing, there must exist coordination between the source node and an intermediate node to avoid looping.

Finally, in future high speed networks, the bottleneck will be on the computation rather on the communication delays. Therefore, it is preferable that all processing intensive functions to be transferred outside of the network to the edges. Source-based protocols satisfy this requirement.

In the next section, we investigate a simple virtual circuit routing problem by simulation.

7 SIMULATION RESULTS

In this section, we investigate a simple case of the derived state dependent virtual circuit routing algorithm via simulation. 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 [18, 53].

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 (Fig. 9) that have the same capacity but the order of their links is different.

Path #1 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 $r * \frac{\gamma}{\delta} = \frac{1000}{700}$, 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 aveage 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.

Fig. 10 and Table 1 describe the simulation results for the network of Fig. 9.

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 1. The average packet delay ± error (95% confidence interval) for the network of Fig. 9, 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.

In this network (Fig. 9), 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*, ortherwise 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 adaptive 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 adaptive 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 transamission speeds μ, N_π packets at path π link and N_p packets at path p link, such that

$$\frac{(1+N_\pi)^2}{\mu} < \frac{(1+N_p)^2}{\mu} \Leftrightarrow \frac{1+2*N_\pi+N_\pi^2}{\mu} < \frac{1+2*N_p+N_p^2}{\mu} \Leftrightarrow$$

$$\Leftrightarrow \frac{(N_\pi - N_p)*(\qquad N_\pi + N_p + 2)}{\mu} < 0 \Leftrightarrow \frac{N_\pi - N_p}{\mu} < 0 \Leftrightarrow$$

$$\Leftrightarrow \frac{N_\pi}{\mu} < \frac{N_p}{\mu} \Leftrightarrow \frac{1+N_\pi}{\mu} < \frac{1+N_p}{\mu}$$

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 adaptive algorithms for a more complex network with unbalanced paths. We consider a network with 5 paths from source to destination.

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 $\frac{r}{\delta} = 20$.

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.

Fig. 12 and Table 2 describe the simulation results of routing 100,000 virtual circuits into the network of Fig. 11.

γ=1/5 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

γ=1/50 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 2. The average packet delay ± error (95% confidence interval) for the network of Fig. 11, for the *Quadratic* and the *Shortest queue* routing with updating every 1, 20, 50, 100 time units.

In this network (Fig. 11), 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*, epecially when the network state information becomes obsolete.

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.

8 CONCLUSIONS

In this paper, 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.

ACKNOWLEDGMENTS

The authors wish to thank Dr. Vikram Saksena of AT & T Bell Laboratories for suggesting the virtual circuit routing problem and for many useful suggestions.

References

[1] M. Athans and P. L. Falb. *Optimal control.* Mc Graw-Hill, 1966.

[2] L. D. Berkovitz. *Optimal control theory.* Springer-Verlag, 1974.

[3] D. P. Bertsekas and R. G. Gallager. *Data Networks.* Prentice Hall, 1987.

[4] 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.

[5] V. G. Boltyanskii. *Mathematical methods of optimal control.* Holt, Rinehart, Winston Inc., 1971.

[6] A.E. Jr. Bryson and Y. C. Ho. *Applied optimal control.* Hemisphere Publ. Co, 1975.

[7] P. J. Courtois and P. Semal. An algorithm for the optimization of nonbifurcated flows in computer communication networks. *Performance Evaluation*, pp. 139-152, 1981.

[8] 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.

[9] 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 Communication, Control, and Computing*, pp. 849-857, Sept. 27-29, 1989.

[10] A.A. Economides, P.A. Ioannou, and J.A. Silvester. Learning automata virtual circuit routing. *EE-Systems Dept, USC*, 1990.

[11] 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*, pp. 613-622, IEEE 1988.

[12] J. Filipiak. Optimal control of store-and-forward networks. *Optimal Control Applications & Methods, Vol. 3, pp. 155-176*, 1982.

[13] J. Filipiak. Dynamic routing in queueing systems with a multiple service facility. *Operations Research, Vol. 32, No. 5, pp. 1163-1180*, 1984.

[14] J. Filipiak. *Modelling and Control of Dynamic Flows in Communication Networks.* Springer-Verlag, 1988.

[15] 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.

[16] W. H. Fleming and R. W. Rishel. *Deterministic and stochastic optimal control.* Springer-Verlag, 1975.

[17] E.M. Gafni. and D.P. Bertsekas. Path assignment for virtual circuit routing. *Proc. ACM SIGCOMM 1983 Conference, pp. 21-25*, ACM 1983.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] M. Gerla, H.W. Chan, and J.R. Boisson De Marca. Routing, flow control and fairness in computer netorks. *Proc. of International Communications Conference, pp. 1272-1275*, IEEE 1984.

[23] M. Gerla and P.O. Nilsson. Routing and flow control interplay in computer networks. *Proc. International Conference on Computer Communications, pp. 84-89*, 1980.

[24] A. Gersht. Analytical model of dynamic routing in virtual circuit packet switching network. *Proc. International Conference on Computer Communications, pp.76-80*, IEEE 1982.

[25] 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.

[26] W. A. Gruver and E. Sachs. *Algorithmic methods in optimal control.* Pitman Advanced Publ., 1980.

[27] J. Y. Hui. Resource allocation for broadband networks. *Journal On Selected Areas In Communications, Vol. JSAC-6*, pp. 1598-1608, 1988.

[28] P.A. Humblet, S.R. Soloway, and B. Steinka. Algorithms for data communication networks-part 2. *Codex, pp. 1-23*, Jun. 1986.

[29] J. M. Jaffe and A. Segall. Threshold design for dynamic routing. *Proc. IEEE International Communications Conference*, pp. 90-92, IEEE 1986.

[30] M. I. Kamien and N. L. Schwartz. Sufficient conditions in optimal control theory. *Journal Economic Theory, Vol 3*, pp. 207-214, 1971.

[31] 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.

[32] D. E. Kirk. *Optimal control theory: an introduction.* Prentice Hall, 1970.

[33] L. Kleinrock. *Queueing Systems, Vol. 1: Theory.* J. Wiley & Sons, 1975.

[34] H. Kobayashi and M. Gerla. Optimal routing in closed queueuing networks. *ACM Transactions on Computer Systems, Vol. 1, No. 4, pp. 294-310*, Nov. 1983.

[35] 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.

[36] 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.

[37] E. B. Lee and L. Markus. *Foundations of optimal control theory.* J. Wiley & Sons, 1967.

[38] Y. S. Lin and J. R. Yee. A distributed routing algorithm for virtual circuit data networks. *Proc. IEEE Infocom Conference*, pp. 200-207, IEEE 1989.

[39] O. L. Mangasarian. Sufficient conditions for the optimal control of nonlinear systems. *Journal SIAM Control, Vol. 4, No. 1*, pp. 139-152, 1966.

[40] 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.

[41] 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.

[42] 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.

[43] A. P. Sage. *Optimum systems control.* Prentice Hall, 1968.

[44] M. Schwartz. *Telecommunication Networks : Protocols, Modeling and Analysis.* Addison Wesley, 1988.

[45] A. Seierstad and K. Sydsaeter. Sufficient conditions in optimal control theory. *International Economic Review, Vol. 18, No. 2*, pp. 367-391, June 1977.

[46] D. Tipper and M.K. Sundareshan. Optimal threshold routing of virtual circuits in data networks. *presented at ORSA/TIMS Joint National Meeting*, 1988.

[47] 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.

[48] W. K. Tsai, P. E. Cantrell, and J. Goos. Fairness of optimal routing in virtual circuit data networks. *Proc. IEEE Infocom*, pp. 119-126, IEEE 1989.

[49] 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.

[50] 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.

[51] J.N. Tsitsiklis and D.P. Bertsekas. Distributed asynchronous optimal routing in data networks. *IEEE Tr. on Automatic Control, Vol. AC-31, No. 4, pp. 325-332*, Apr. 1986.

[52] P.P. Varaiya. *Notes on optimization.* Van Nostrand Reinhold Co., 1972.

[53] E. F. Wunderlich. An analysis of dynamic virtual circuit routing. *Proc. IEEE National Telecommunications Conference*, pp. A3.3. - A3.3.6, IEEE 1981.

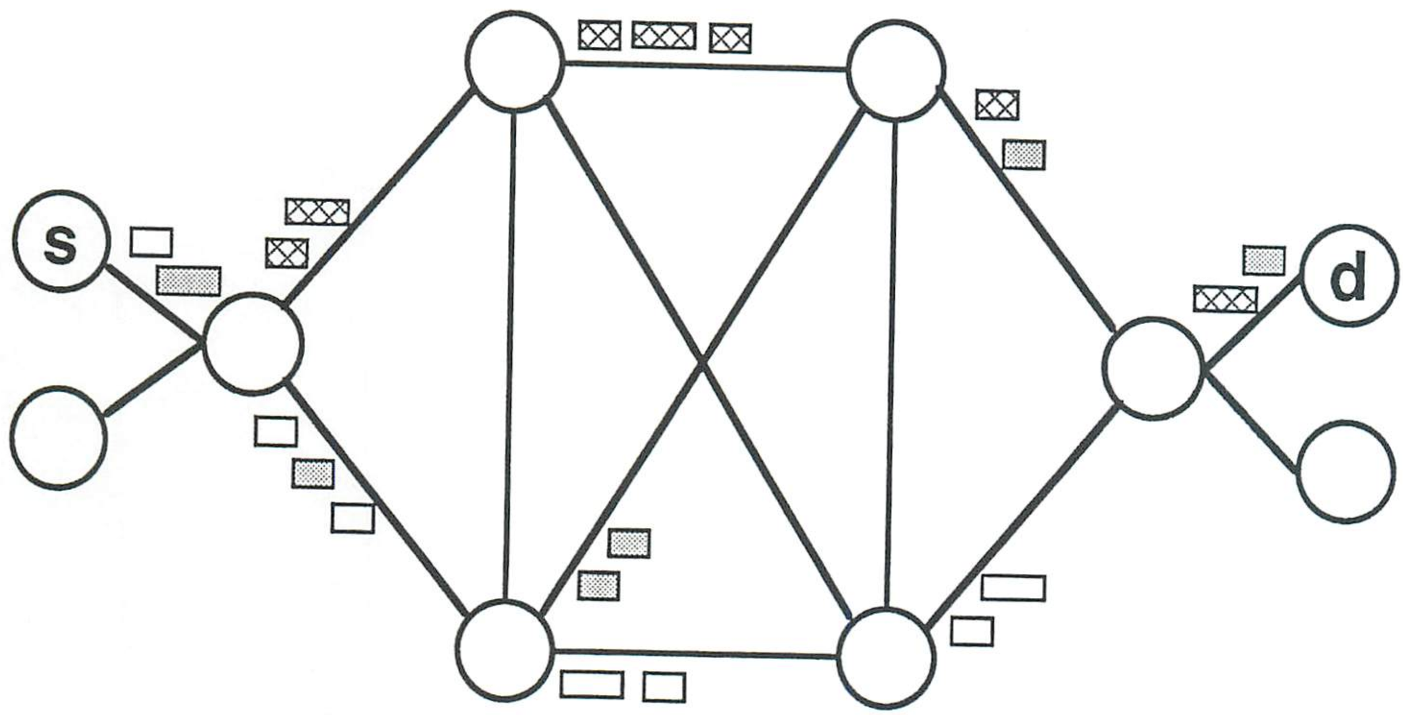


Fig. 1 A virtual circuit network

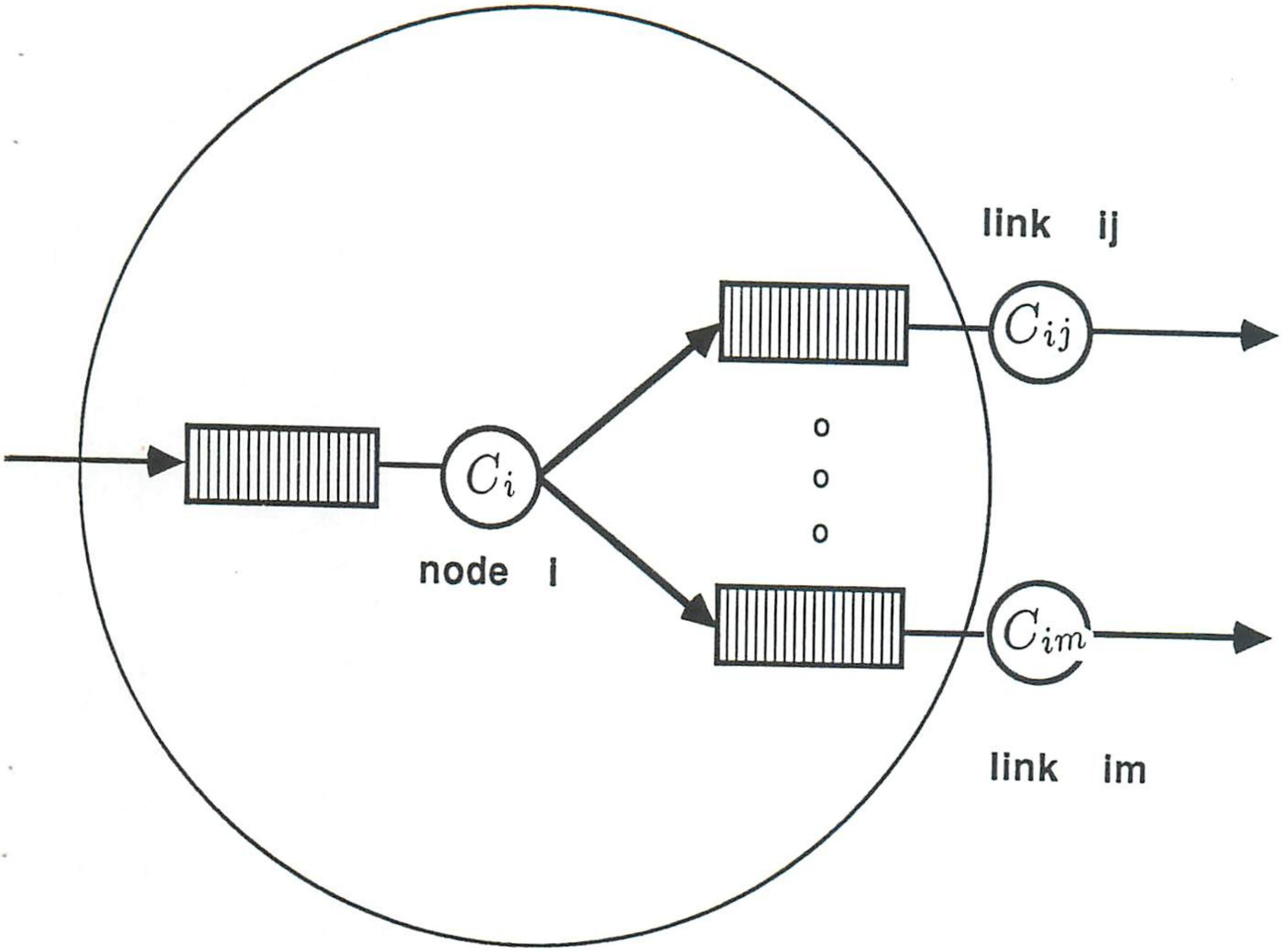


Fig. 2 A network node

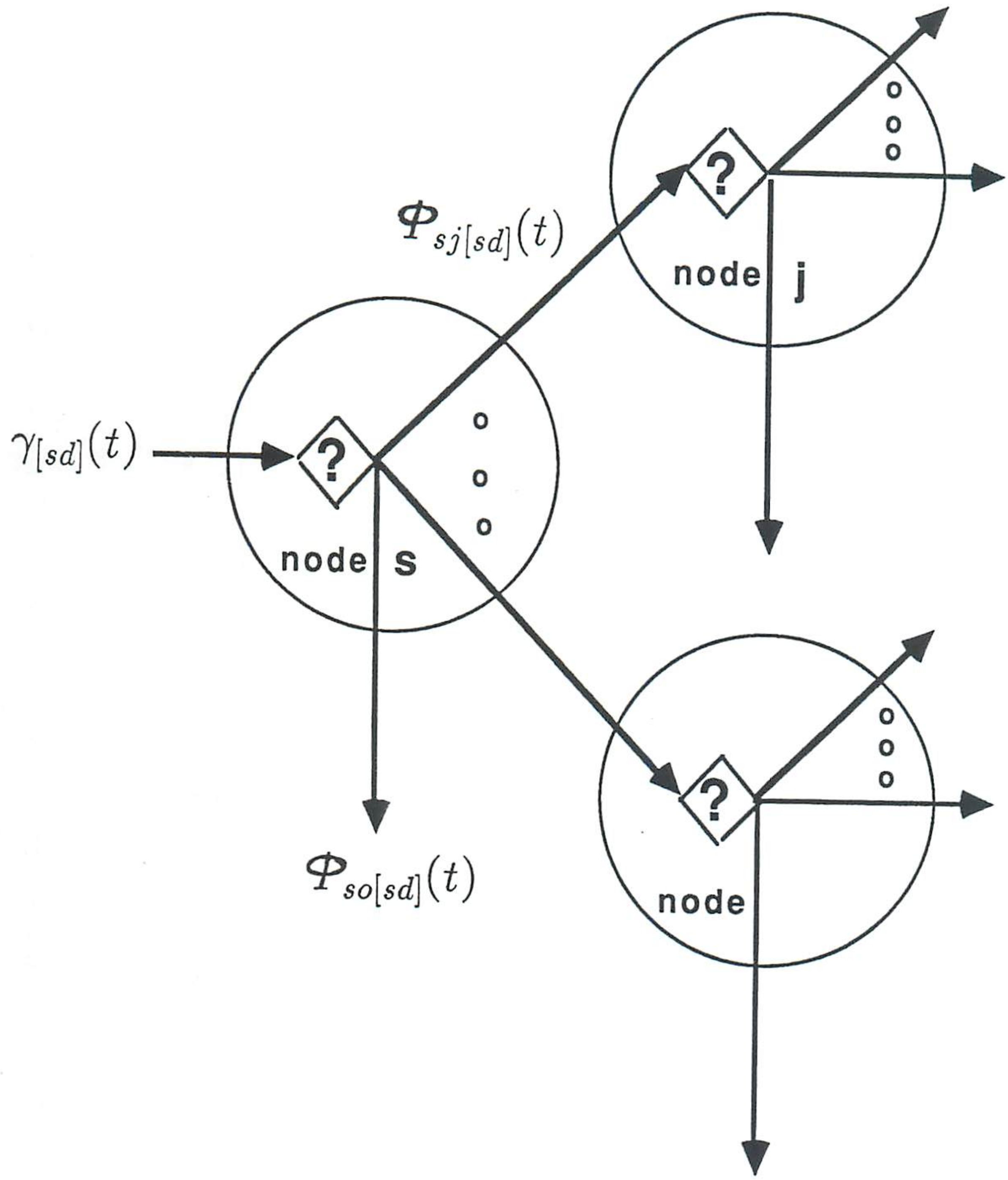


Fig. 3 Virtual circuit congestion control and routing

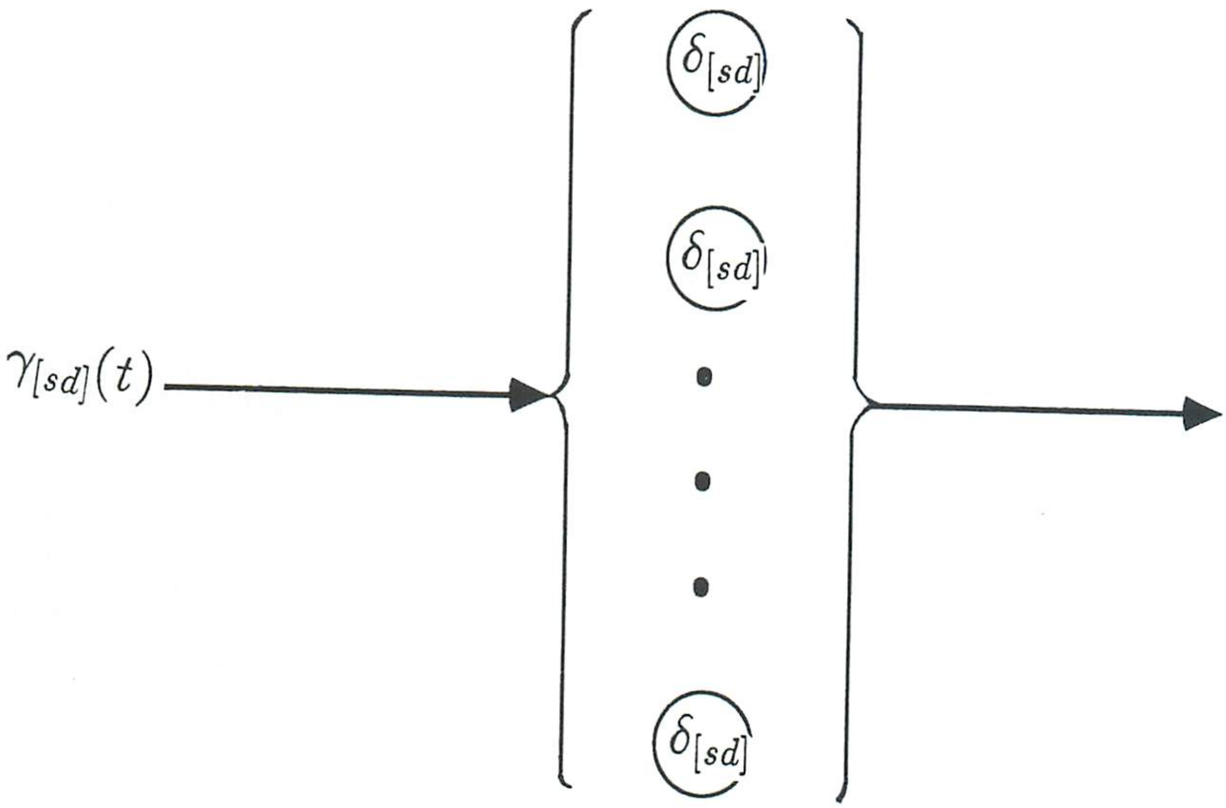


Fig. 4 $M/M/\infty$ model for virtual circuit process

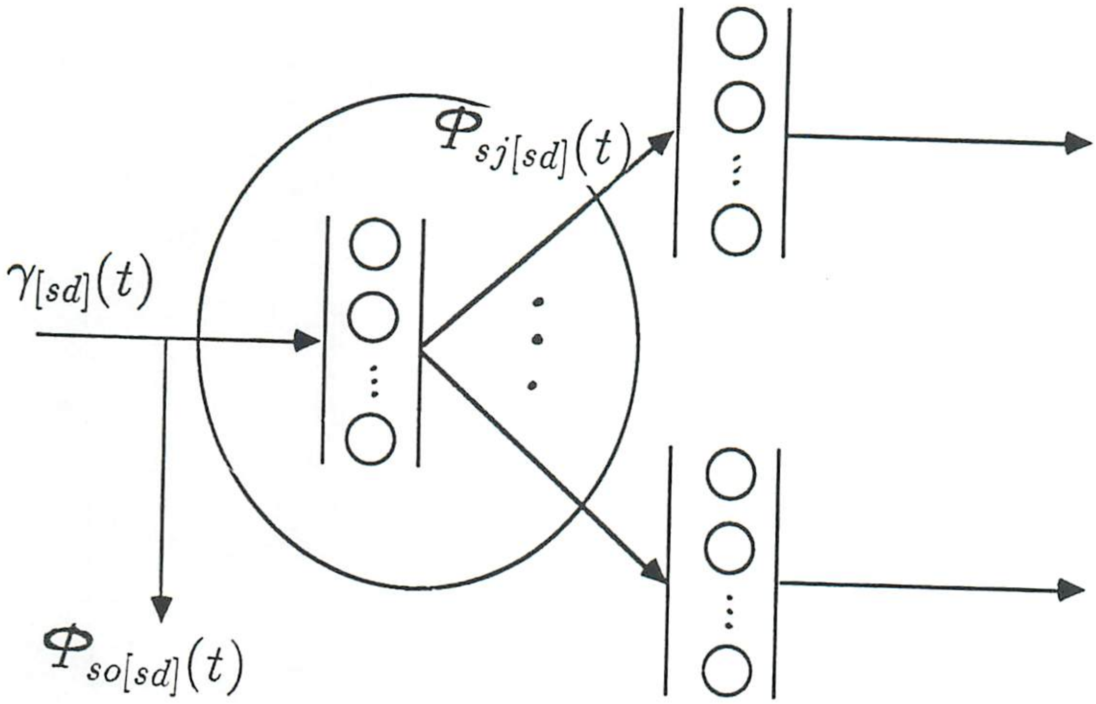


Fig. 5 Virtual circuit processes

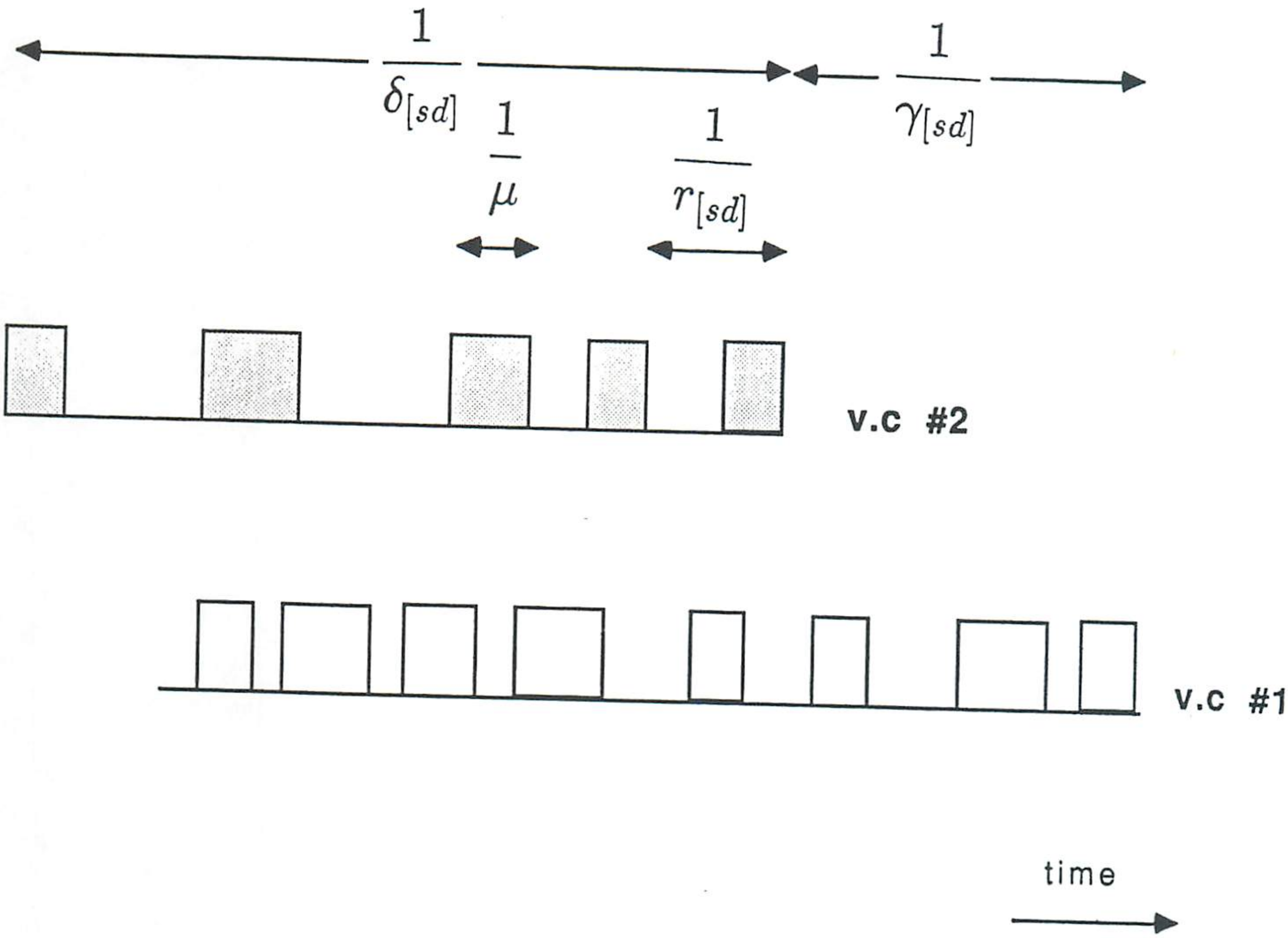


Fig. 6 Two virtual circuits and their packets

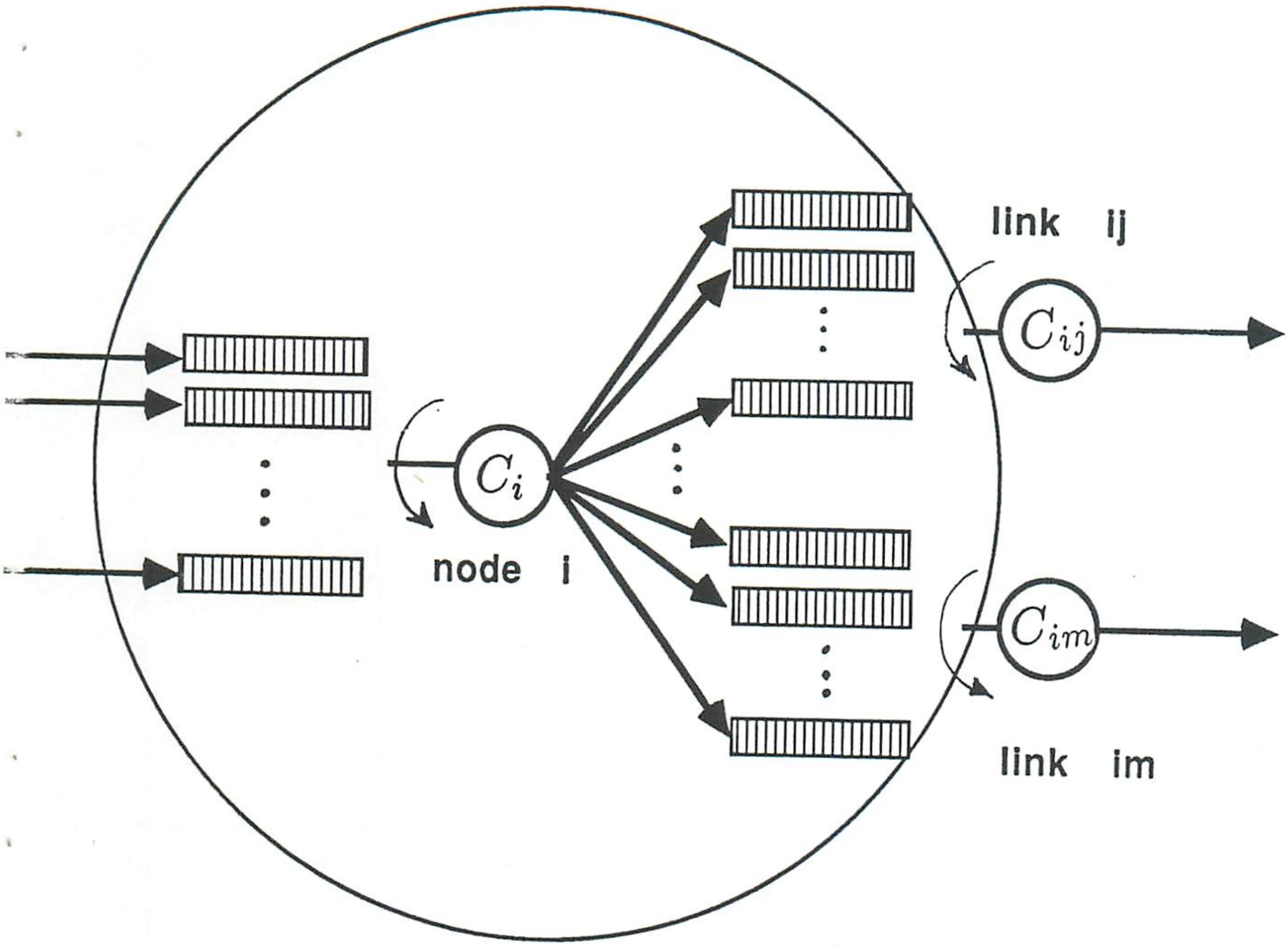


Fig. 8 Processor sharing model for packet process

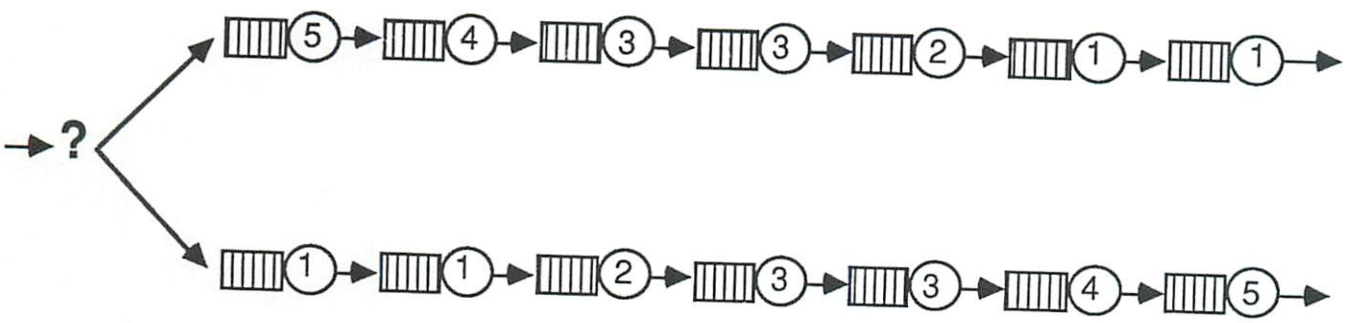
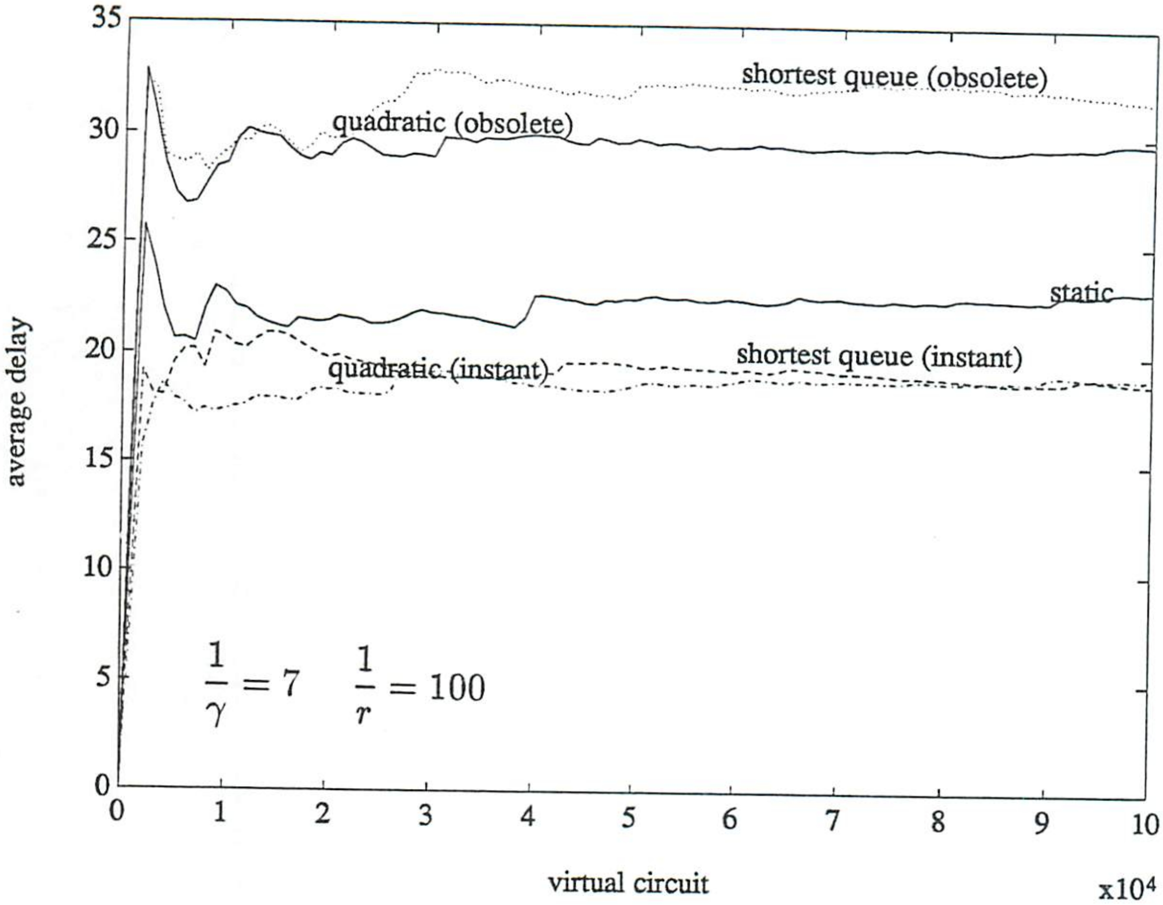
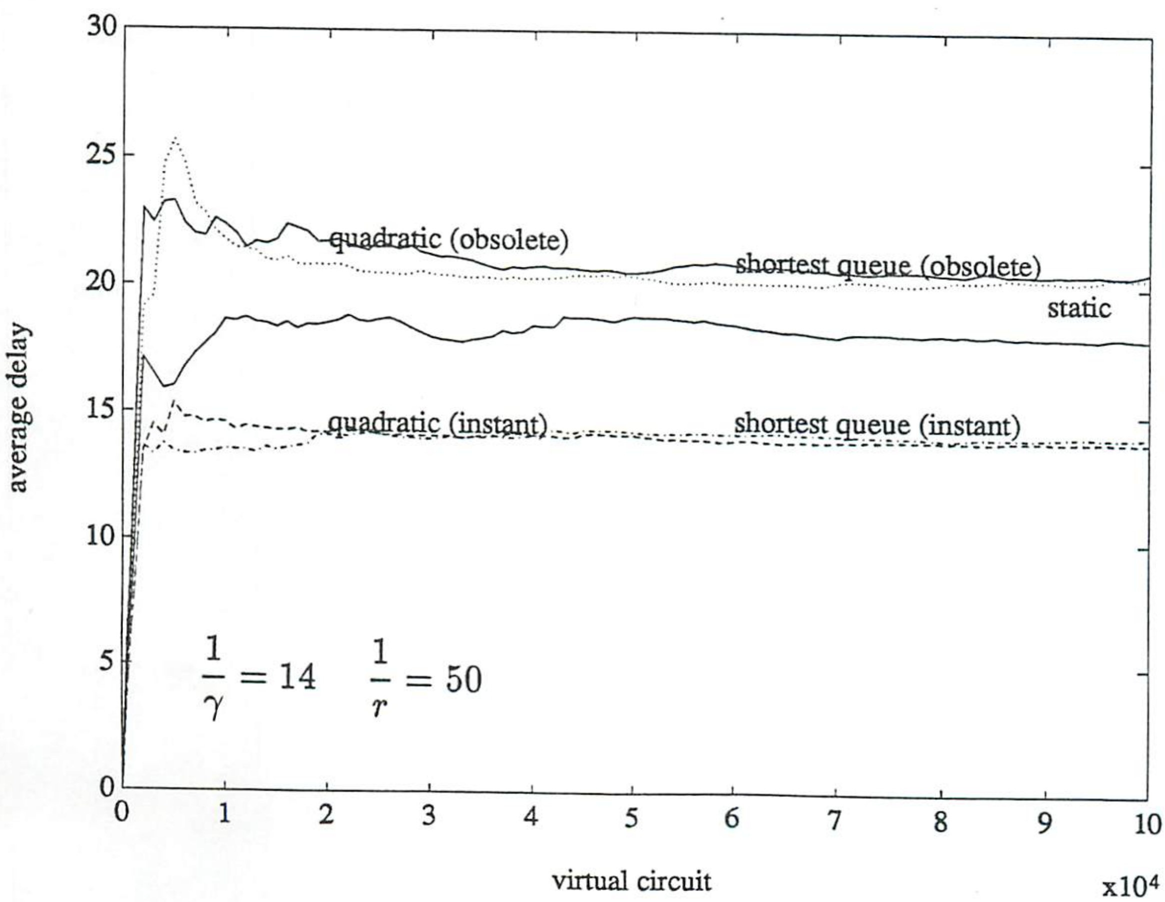


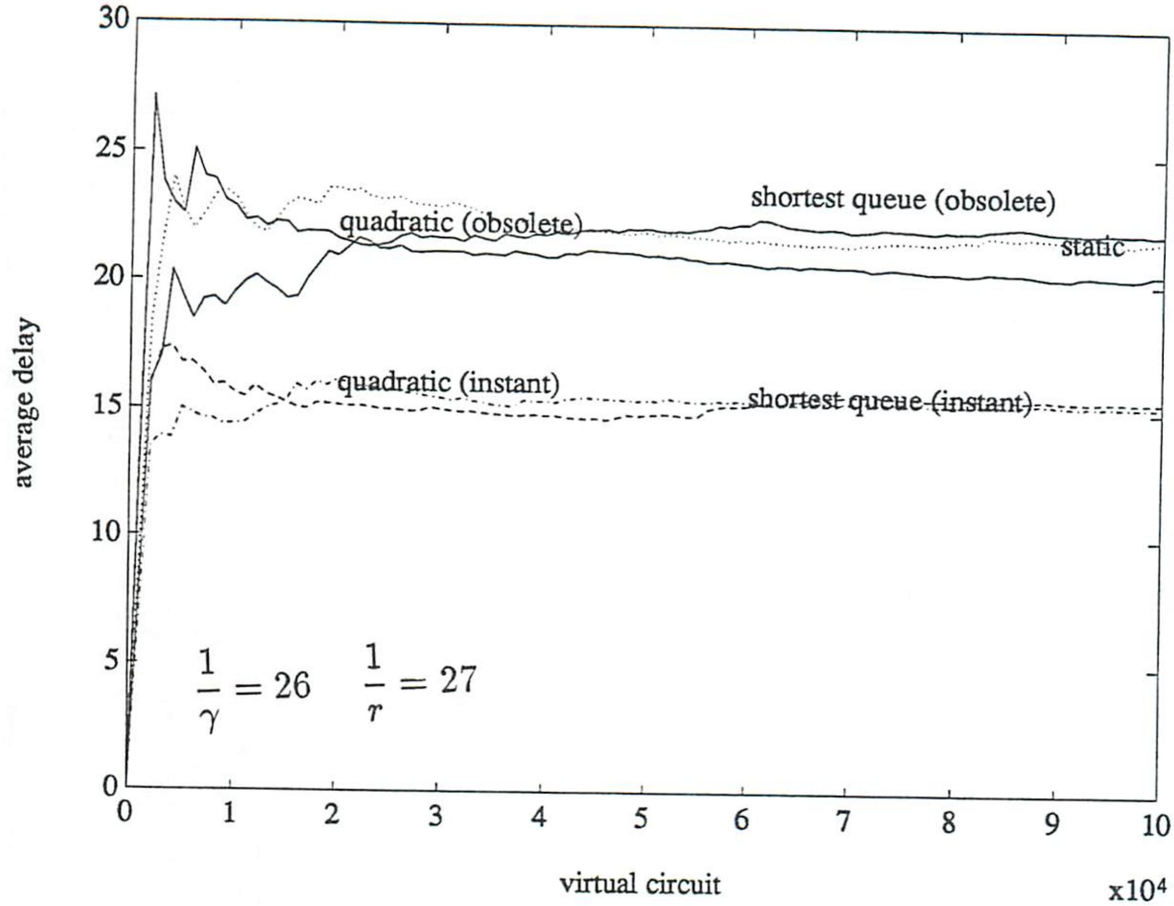
Fig. 9 Simulated network



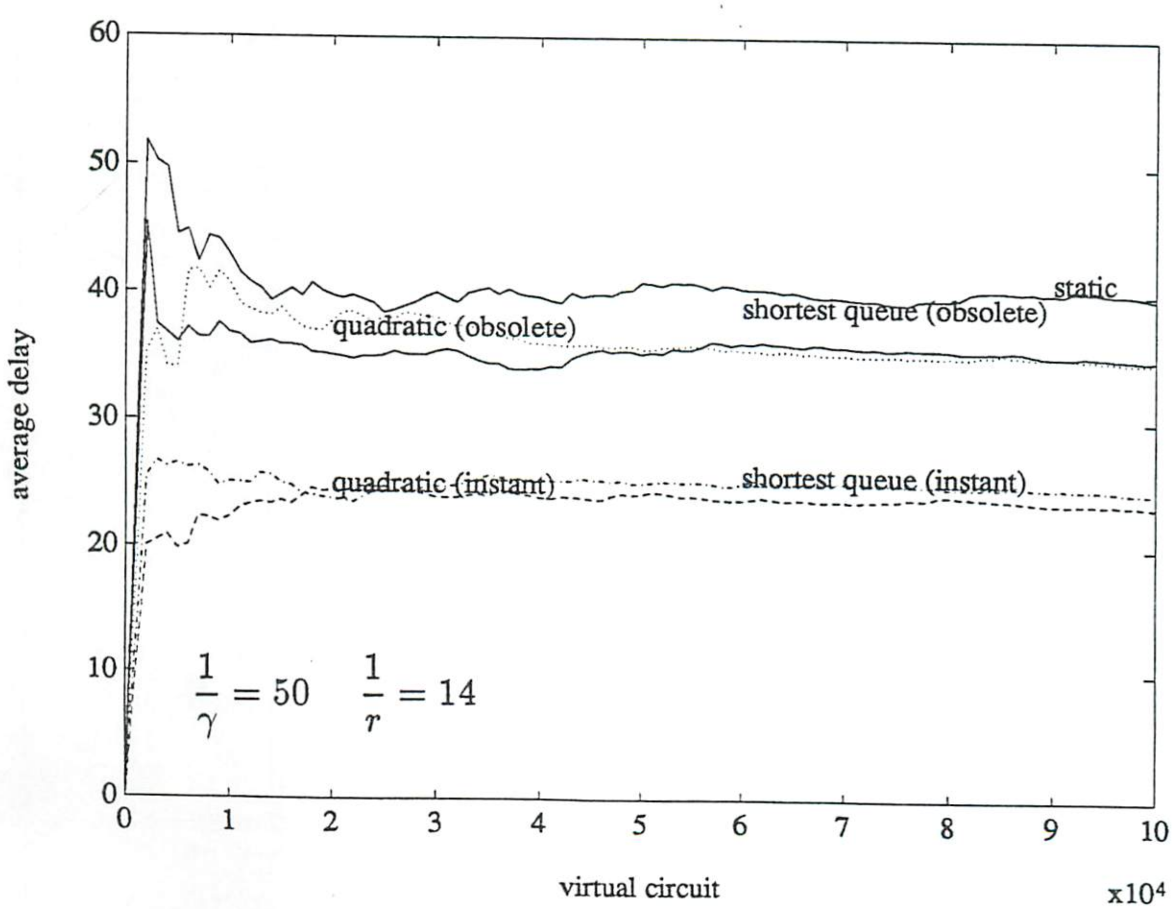
shortest queue (obsolete)
quadratic (obsolete)
static
quadratic (instant)
shortest queue (instant)
average delay
virtual circuit
x10^4
$\frac{1}{\gamma} = 7 \quad \frac{1}{r} = 100$



quadratic (obsolete)
shortest queue (obsolete)
static
quadratic (instant)
shortest queue (instant)
average delay
virtual circuit
x10^4
$\frac{1}{\gamma} = 14 \quad \frac{1}{r} = 50$



30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10
average delay
virtual circuit
x10⁴
shortest queue (obsolete)
quadratic (obsolete)
static
quadratic (instant)
shortest queue (instant)
$\frac{1}{\gamma} = 26 \quad \frac{1}{r} = 27$



60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
9
10
average delay
virtual circuit
x10⁴
static
shortest queue (obsolete)
quadratic (obsolete)
quadratic (instant)
shortest queue (instant)
$\frac{1}{\gamma} = 50 \quad \frac{1}{r} = 14$

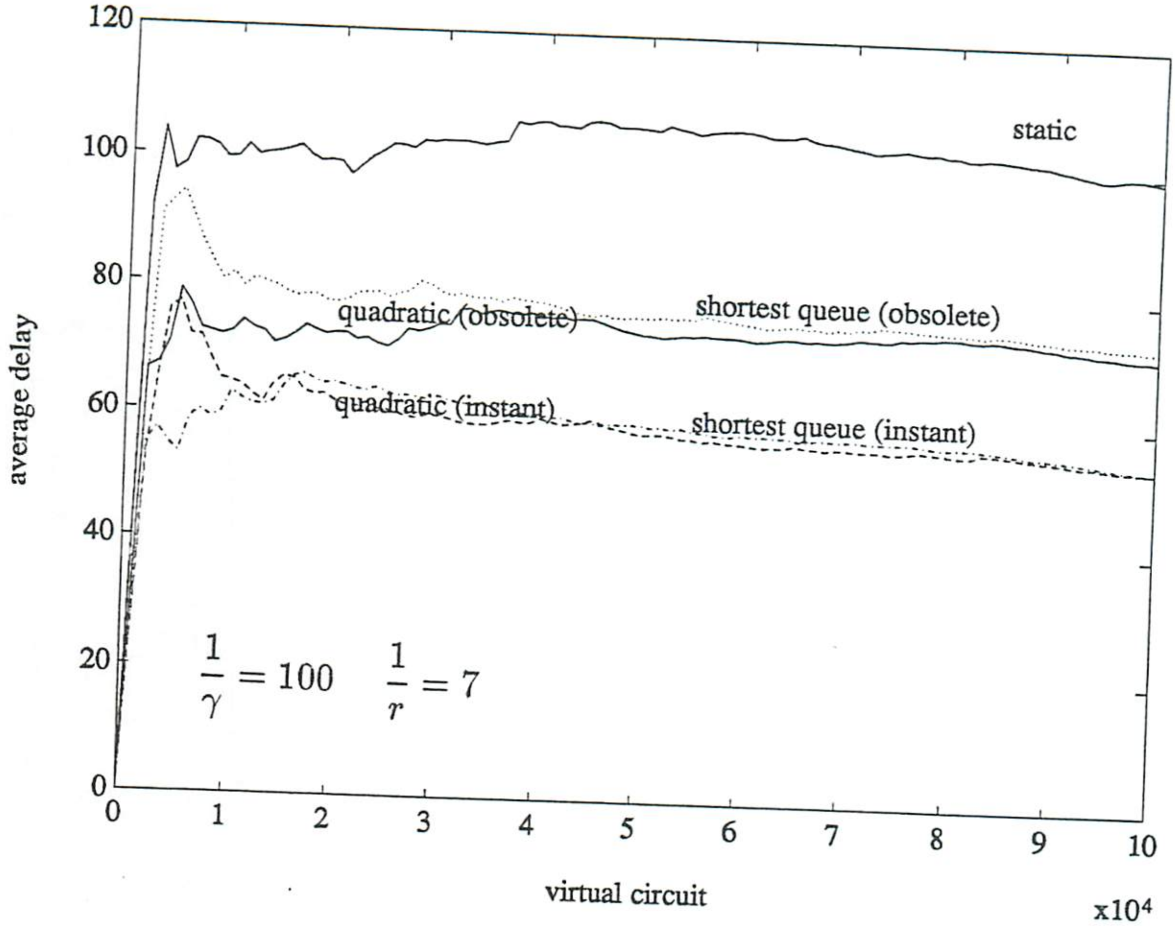


Fig. 10 The average packet delay for the network of Fig. 9, for the Quadratic routing with instantaneous and obsolete information, the Shortest queue routing with instantaneous and obsolete information and the Optimal static routing implemented as Round-Robin.

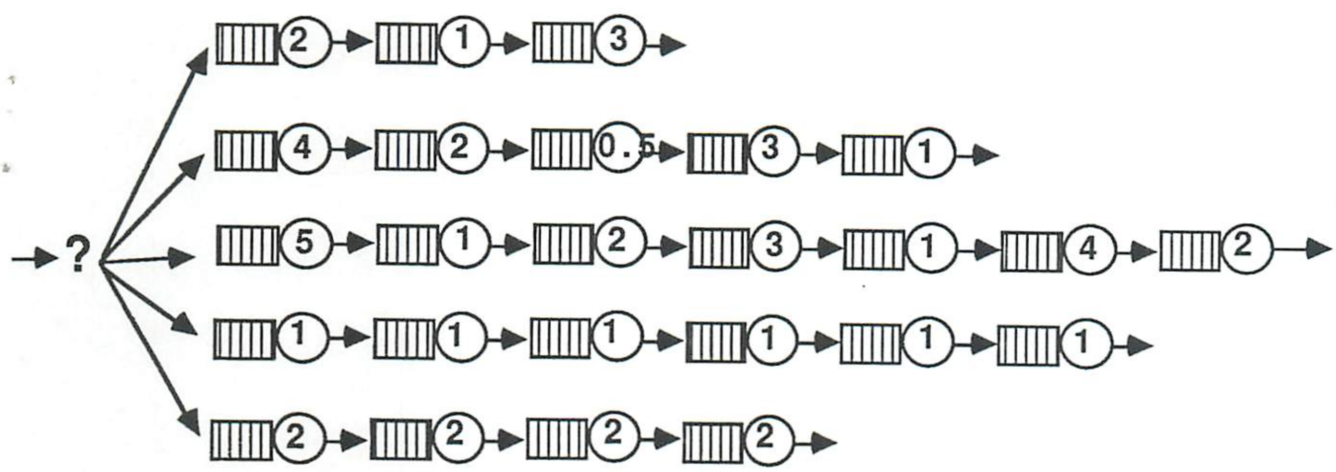
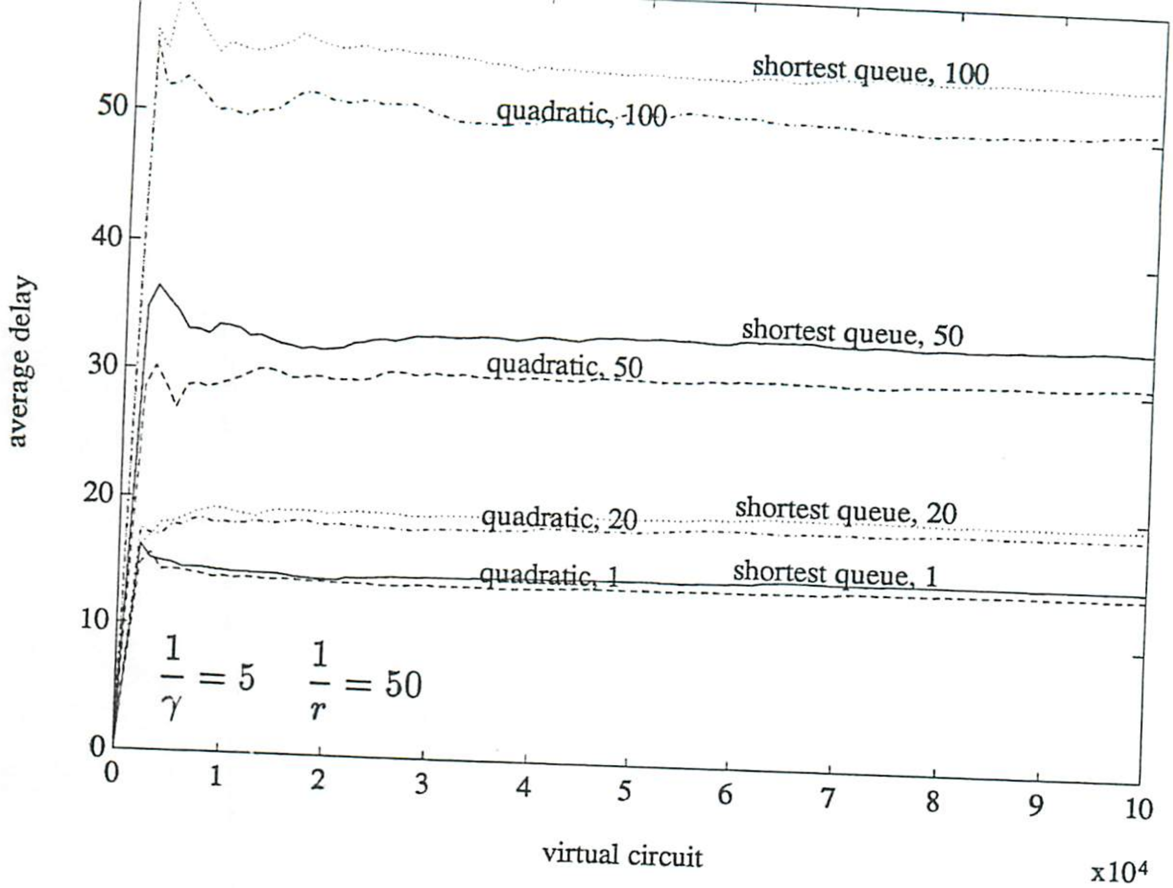


Fig. 11 Simulated network



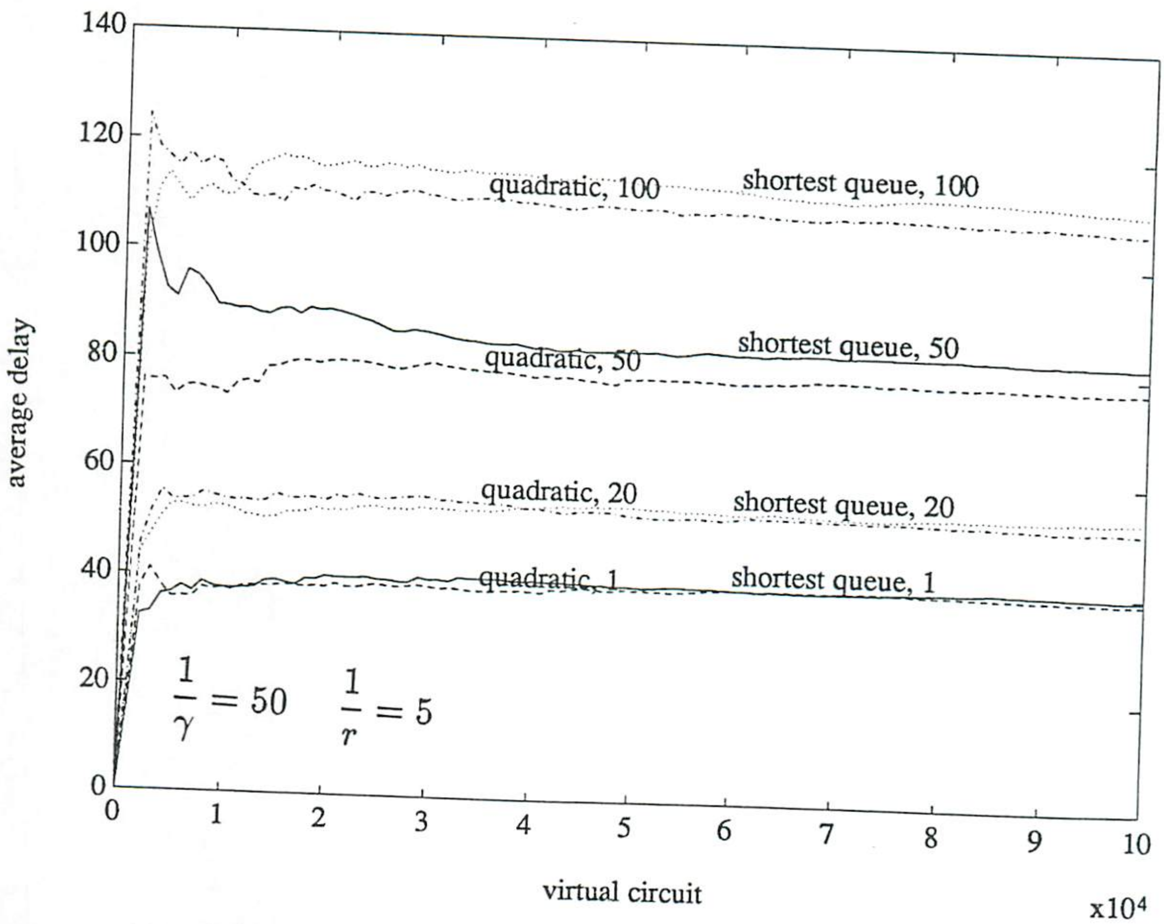


Fig. 12 The average packet delay for the network of Fig. 11, for the Quadratic routing with updating every 1, 20, 50, 100 time units and the Shortest queue routing with updating every 1, 20, 50, 100 time units.