Economides, A.A., Ioannou, P.A. & Silvester, J.A.: **Adaptive virtual circuit routing**. *Computer Networks and ISDN Systems*, Vol. 28, No 1, pp. 1-8, 1995

ELSEVIER

Computer Networks and ISDN Systems 28 (1996) 401-409

COMPUTER NETWORKS and ISDN SYSTEMS

Adaptive virtual circuit routing

A.A. Economides [a,*], P.A. Ioannou [b], J.A. Silvester [b]

[a] *University of Macedonia, Thessaloniki 54006, Greece*

[b] *University of Southern California, Los Angeles, CA 90089-2562, USA*

Accepted 12 January 1995

Abstract

The paper deals with the adaptive routing problem in virtual circuit communication networks. A newly arriving call at a source node is routed along the minimum length path to its destination node. All packets belonging to this call follow the same path through the network.

The superiority of a quadratic state-dependent routing algorithm to the shortest-queue routing algorithm is shown via simulation. The sooner the network state information becomes available to the router and the more often that this information is updated, the smaller the achieved average packet delay. Also, the age of this information at the router should be less or at least comparable to (but not extremely larger than) the mean interarrival time of virtual circuits.

Keywords: Adaptive routing; Connection-oriented; Dynamic routing; Simulation; State-dependent routing; Virtual channel; Virtual circuit; Virtual connection; Virtual route

1. Introduction

Many public data networks (Euronet, Telenet, Transpac, Tymnet) adopt the X.25 protocol that requires the network to set up a virtual circuit for each user call or session. In virtual circuit networks, for each call a virtual circuit is established along a single path 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 be 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 [10].

* Corresponding author. Fax +30 31-846322, E-mail: economid@macedonia.uom.gr.

SSDI 0169-7552(95)00006-2

(7) Easier admission 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.

There are three types of virtual circuit services: permanent virtual circuit, switched virtual circuit (or virtual call) and fast select. Permanent virtual circuit service provides an pre-established connection between the end points. So, it does not need to set up and disconnect the connection [9]. Switched virtual circuit service requires a connection to be established before data transfer starts. So, there is also a set up delay. In fast select service the control packet that sets up the connection carries data as well [8]. So, there is no set up delay. In all three virtual circuit services, the routing decision is done when the first packet of the virtual circuit arrives. Subsequently, all other packets belonging to this virtual circuit follow the same route.

Routing (defined at the Network layer of OSI) is the selection of the best path through the network from source to destination. In adaptive (or dynamic) routing, the routing decisions depend on the current network state, and are done independently for each virtual circuit. In a real virtual circuit network, there are two general ways to select a route:

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

(2) In *probabilistic routing*, each virtual circuit is routed along the minimum length path with high probability. Note that the deterministic routing may be considered as a special case of the probabilistic routing (with probability 1). One possible way to update the routing probabilities is using learning automata [2].

Furthermore, the routing decisions may be done either at the source node or at every intermediate node to the destination:

(1) In *source routing*, each source node routes a newly arriving virtual circuit at this source node along the current minimum length path to the destination node.

(2) In *node-by-node routing* (or *link-by-link routing* [11]), each network node routes a newly arriving virtual circuit at this node to the next node along the current minimum length path to the destination node. Thus, it provides complete decentralization.

In node-by-node routing, every node needs to keep the information about all paths from this node to all destinations. However, it also needs to know which path the virtual circuit has followed to reach this node in order to avoid looping. Therefore, there must exist coordination between the source node and an intermediate node to avoid looping. In source routing every source node needs to keep information about all paths from this source node to all destinations. In this case, it is also trivial to guarantee no looping. 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. Finally, in future high speed networks, the bottleneck will be on the computation rather than 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.

The adaptive (or dynamic) routing problem in virtual circuit networks has been analyzed in [4,3,5-7,12]. However, only [2,11] use simulation to validate their results for a real network implementation. In this paper, we compare via simulation two adaptive deterministic source routing algorithms in virtual circuit networks. We show the superiority of a quadratic state-dependent routing algorithm to the shortest-queue routing algorithm.

2. Simulation

Consider a virtual circuit network, where calls arrive (Poisson) at a rate γ and their duration (exponential) is on the average $1/\delta$. The packets that belong to a given call arrive (Poisson) at a rate r and the packet service requirement (exponential) has mean $1/\mu$. Let N_{ij} be the number of packets at link ij and C_{ij} the transmission rate at link ij.

Using optimal control theory, we can express the length "seen" by a newly arriving virtual circuit on a link as the sum of two terms [4]. The first term is the cost/time-unit for maintaining this virtual circuit at this link *times* the average virtual circuit duration. The second term is the average number of packets in this virtual circuit *times* (the cost/time-unit for a packet *times* a quadratic function of the average number of packets at this link *minus* the profit from servicing a packet). Next, we investigate this link length via simulation. We concentrate on the effect of the quadratic function of the average number of packets at this link.

We compare via simulation two adaptive virtual circuit routing algorithms. They route every new virtual circuit along the minimum length path (ties are broken arbitrarily—though we seldom have ties). The first algorithm uses as link length a quadratic function of the average number of packets at this link. The second algorithm uses as link length the average packet delay on this link.

(1) *quadratic routing*: send a new virtual circuit along path π, if

$$\sum_{ij\in\pi} \frac{(1+N_{ij})^2}{\mu_{ij}} = \min_{p} \left\{ \sum_{ij\in p} \frac{(1+N_{ij})^2}{\mu_{ij}} \right\}.$$

(2) *shortest-queue routing*: send a new virtual circuit along path π, if

$$\sum_{ij\in\pi} \frac{1+N_{ij}}{\mu_{ij}} = \min_{p} \left\{ \sum_{ij\in p} \frac{1+N_{ij}}{\mu_{ij}} \right\}.$$

For updating the information at the source node about the link lengths in the network, we consider 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 [12].

(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.

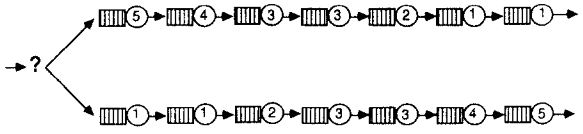

Fig. 1. Two-path network.

2.1. Balanced network

First, we consider a single source-destination network with 2 paths from source to destination (Fig. 1) 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 $1/\mu = 1$ and therefore $\mu_{ij} = \mu * C_{ij} = C_{ij}$. The mean virtual circuit duration is $1/\delta = 1000$. The total packet arrival rate is $r * \gamma/\delta = \frac{1000}{700}$, however we consider 5 cases that achieve this total packet arrival rate:

γ	r	γ/δ	r/δ
$\frac{1}{7}$	$\frac{1}{100}$	$\frac{1000}{7}$	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, γ/δ is the average number of virtual circuits into the network and r/δ 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:

(1) *instantaneous* information. At every instant, the source node knows the current number of packets at every link.

(2) *obsolete* information. 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. This information is used by the source node after 50 time units delay.

Figure 2 and Table 1 describe the simulation results of routing 100 000 virtual circuits into the network of

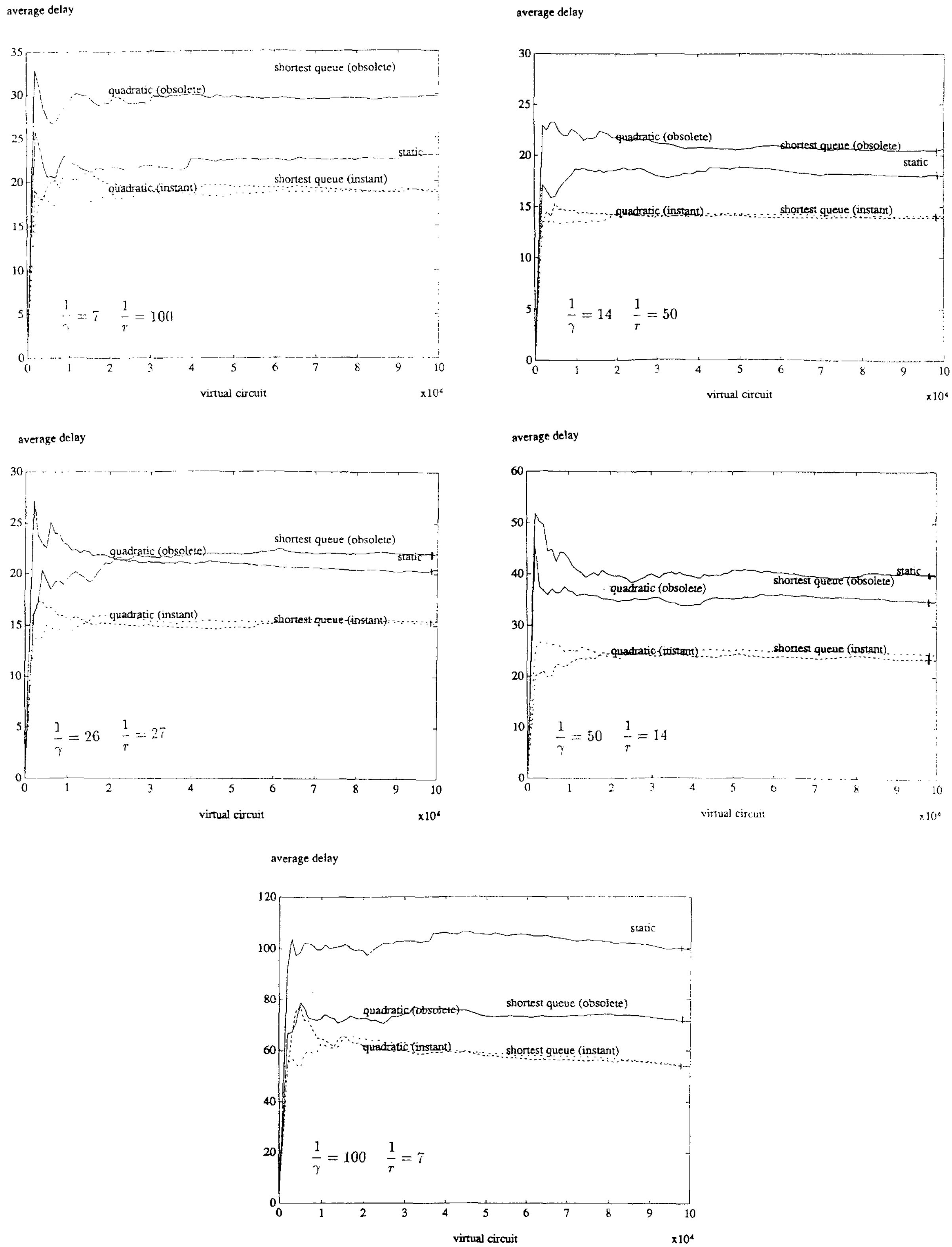

Fig. 2. The average packet delay for the network of Fig. 1.

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

	Instantaneous	Obsolete
$\gamma=1/7$, $r=1/100$		
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	
$\gamma=1/14$, $r=1/50$		
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	
$\gamma=1/26$, $r=1/27$		
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	
$\gamma=1/50$, $r=1/14$		
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	
$\gamma=1/100$, $r=1/7$		
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	

Fig. 1. Both paths receive on the average the same number of virtual circuits and have the same average packet delay, since they have similar links but in different positions.

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 the last case ($\gamma = 1/100, r = 1/7$), where each virtual circuit carries a large number of packets. This may happen because a wrong routing decision for a heavily loaded virtual circuit seriously affects the load on the selected path. In contrast, a wrong routing decision for a lightly loaded virtual circuit does not seriously affect the load on the selected path.

The more often that we update the link length information at the source node, the smaller the achieved average packet delay. The sooner the link length information becomes available to the source node, the smaller the achieved average packet delay. 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.

For comparison reasons, we also show the average packet delay when we split the virtual circuit traffic to the two paths. In a round-robin basis an odd numbered virtual circuit is routed to path #1 and an even numbered virtual circuit is routed to path #2. This may be considered as an implementation of the *optimal quasi-static routing*.

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 quasi-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 a 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

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

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

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

$$\Leftrightarrow \frac{N_\pi - N_p}{\mu} < 0$$

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

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

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

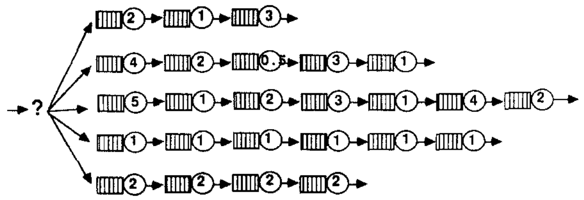

Fig. 3. Five-path network.

number of packets at their links satisfying the following relations simultaneously:

$$\sum_{ij\in\pi} \frac{(1+N_{ij})^2}{\mu_{ij}} < \sum_{xy\in p} \frac{(1+N_{xy})^2}{\mu_{xy}},$$

$$\sum_{xy\in p} \frac{1+N_{xy}}{\mu_{xy}} < \sum_{ij\in\pi} \frac{1+N_{ij}}{\mu_{ij}}.$$

Then the *quadratic routing* will choose path π, while the *shortest-queue routing* will choose path p.

2.2. Unbalanced network

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 (Fig. 3). 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 #4 has 6 links with transmission speeds 1, 1, 1, 1, 1 and 1. Path #5 has 4 links with transmission speeds 2, 2, 2 and 2.

The mean packet service time is $1/\mu = 1$ and therefore $\mu_{ij} = \mu * C_{ij} = C_{ij}$. The mean virtual circuit duration is $1/\delta = 1000$. 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 $\gamma/\delta = 200$ and the average number of packets per virtual circuit is $r/\delta = 20$.

In case #2, 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 $\gamma/\delta = 20$ and the average number of packets per virtual circuit is $r/\delta = 200$.

The information at the source node about the link lengths in the network is updated according to four schemes:

(1) *1 time unit.* At every instant, the source node knows the current number of packets at every link.

(2) *20 time units.* 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. This information is used by the source node after 20 time units delay.

(3) *50 time units.* 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. This information is used by the source node after 50 time units delay.

(4) *100 time units.* 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. This information is used by the source node after 50 time units delay.

Figure 4 and Table 2 describe the simulation results of routing 100 000 virtual circuits into the network of Fig. 3.

In this network (Fig. 3), 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 sooner the link length information becomes available to the source node, the smaller the achieved average packet delay. However, the *quadratic routing* achieves clearly smaller average packet delay than the *shortest-queue routing*, especially 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 confirms our previous observation that for virtual circuit networks it is not enough to consider the aggregate packet arrival process, but both the virtual circuit and packets per virtual circuit processes.

Table 2
The average packet delay ± error (95% confidence interval) for the network of Fig. 3, for the *quadratic* and the *shortest-queue* routing with updating every 1, 20, 50, 100 time units

	1 time	20 time	50 time	100 time
$\gamma=1/5$, $r=1/50$				
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
$\gamma=1/50$, $r=1/5$				
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

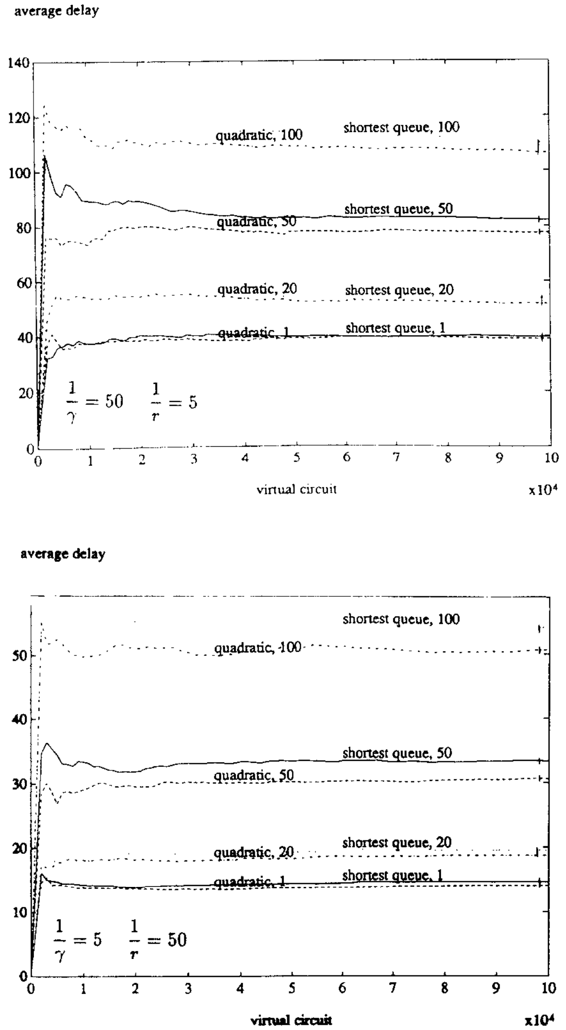

Fig. 4. The average packet delay for the network of Fig. 3.

3. Conclusions

We compare via simulation two adaptive deterministic source routing algorithms in virtual circuit networks. We demonstrate, that for an unbalanced network, the *quadratic routing* achieves smaller average packet delay than the *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*.

We find some interesting results about the update frequency and the delay after which the link length information is available at the router:

The more often that we update the link length information at the router, the smaller the achieved average packet delay.

The sooner the link length information becomes available to the router, the smaller the achieved average packet delay.

Also, the age of the network state information at the router should be less or at least comparable to (but not extremely larger than) the mean interarrival time of virtual circuits. Otherwise, the router depends on out-of-date network state information and makes wrong routing decisions.

Finally, for different traffic scenaria, we note that even if the total packet arrival rate is the same, the average packet delay may be extremely different. Thus, for virtual circuit networks it is not enough to consider the aggregate packet arrival process, but both the virtual circuit and packets per virtual circuit processes.

Acknowledgements

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

References

[1] K.-J. Chen and K.Y. Ho and V.R. Saksena, Analysis and design of a highly reliable transport architecture for ISDN frame-relay networks, *IEEE Journal on Selected Areas in Communications* 7 (8) (1989) 1231-1242.

[2] A.A. Economides and P.A. Ioannou and J.A. Silvester, Decentralized adaptive routing for virtual circuit networks using stochastic learning automata, *Proc. IEEE Infocom 88 Conf.*, 1988, pp. 613–622.

[3] A.A. Economides and P.A. Ioannou and J.A. Silvester, Adaptive routing and congestion control for window flow controlled virtual circuit networks, *Proc. 27th Annual Allerton Conf. on Communications, Control, and Computing*, September 1989, pp. 849–857.

[4] A.A. Economides and P.A. Ioannou and J.A. Silvester, Dynamic routing and admission control for virtual circuit networks, *Journal of Network and Systems Management*, **3** (2) (1995) 173–194.

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

[6] A. Gersht, Analytical model of dynamic routing in virtual circuit packet switching network, *Proc. Int. Conf. on Computer Communications*, 1982, pp.76–80.

[7] J.M. Jaffe and A. Segall, Threshold design for dynamic routing, *Proc. IEEE Int. Communications Conf.*, 1986, pp. 90–92.

[8] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison-Wesley, Reading , Mass. (1987).

[9] J.D. Spragins and J.L. Hammond and K. Pawlikowski, *Telecommunications Protocols and Design*, Addison-Wesley, Reading, Mass. (1991).

[10] W. Stallings, *ISDN and Broadband ISDN*, Macmillan, New York (1992).

[11] D. Tipper and M.K. Sundareshan, An optimal control approach to decentralized dynamic virtual circuit routing in computer networks, *Proc. IEEE Infocom 90*, 1990, pp. 926–933.

[12] E.F. Wunderlich, An analysis of dynamic virtual circuit routing, *Proc. IEEE Nat. Telecommunications Conf.*, 1981, pp. A3.3.– A3.3.6.

Anastasios A. Economides was born and grew up in Thessaloniki, Greece. He received the Diploma degree in Electrical Engineering from Aristotle University of Thessaloniki, in 1984. After receiving a Fulbright and a Greek State Fellowship, he continued for graduate studies at the United States. He received a M.Sc. and a Ph.D. degree in Computer Engineering from the University of Southern California, Los Angeles, in 1987 and 1990, respectively. During his graduate studies, he was a research assistant performing research on routing and congestion control. He is currently an Assistant Professor of Informatics at the University of Macedonia, Thessaloniki. His research interests are in the area of Performance Modeling, Optimization and Control of High-Speed Networks. He is a member of IEEE, ACM, INFORMS, EPY, TEE.

Petros A. Ioannou received the B.Sc. degree with First Class Honors from University College, London, England, in 1978 and the M.S. and Ph.D. degrees from the University of Illinois, Urbana, Illinois, in 1980 and 1982, repectively. In 1982, Dr. Ioannou joined the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, California. He is currently a Professor in the same Department and the Director of the Center of Advanced Transportation Technologies. He teaches and conducts research in the areas of adaptive control, neural networks and intelligent vehicle and highway systems. He is the recipient of the 1984 outstanding IEEE Transactions Paper Award and the 1985 Presidential Young Investigator Award. He has been an associate Editor for the IEEE Transactions on Automatic Control from 1987 to 1990 and he is currently on the Editorial Board for the International Journal of Control and Automatica. Dr. Ioannou is a fellow of IEEE, a member of SAE, IVHS America, and of the AVCS Committee of IVHS America.

John A. Silvester was born in England and grew up in London. He received an MA in Mathematics and Operations Research from Cambridge University. He emigrated to the United States and studied at West Virginia University for an MS in Statistics and Operations Research. The research topic of his Ph.D. in Computer Science from UCLA was Packet Radio Networks. Since 1979 he has been on the faculty at the University of Southern California where he is a Professor of

Electrical Engineering and Vice-Provost for Academic Computing. He has been active in the IEEE Communications Society and was Chair of the First Workshop on Computer Communications (1986) and General Chair of Infocom (1990). He has written over 100 research papers on various performance modelling and design aspects of computer communication networks.