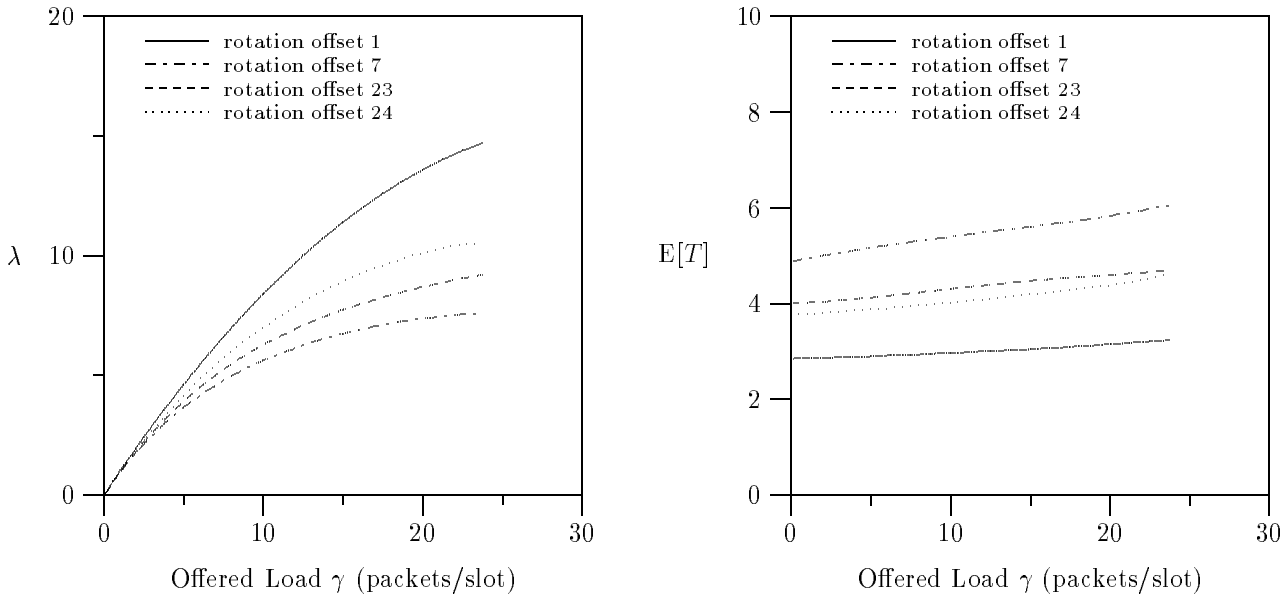


Figure 13: Predicted throughput λ (in packets per time slot) and delay E[T] (in time slots) versus scaled offered load γ in a single-buffer 5×5 Manhattan network under nonuniform, permutation traffic loads

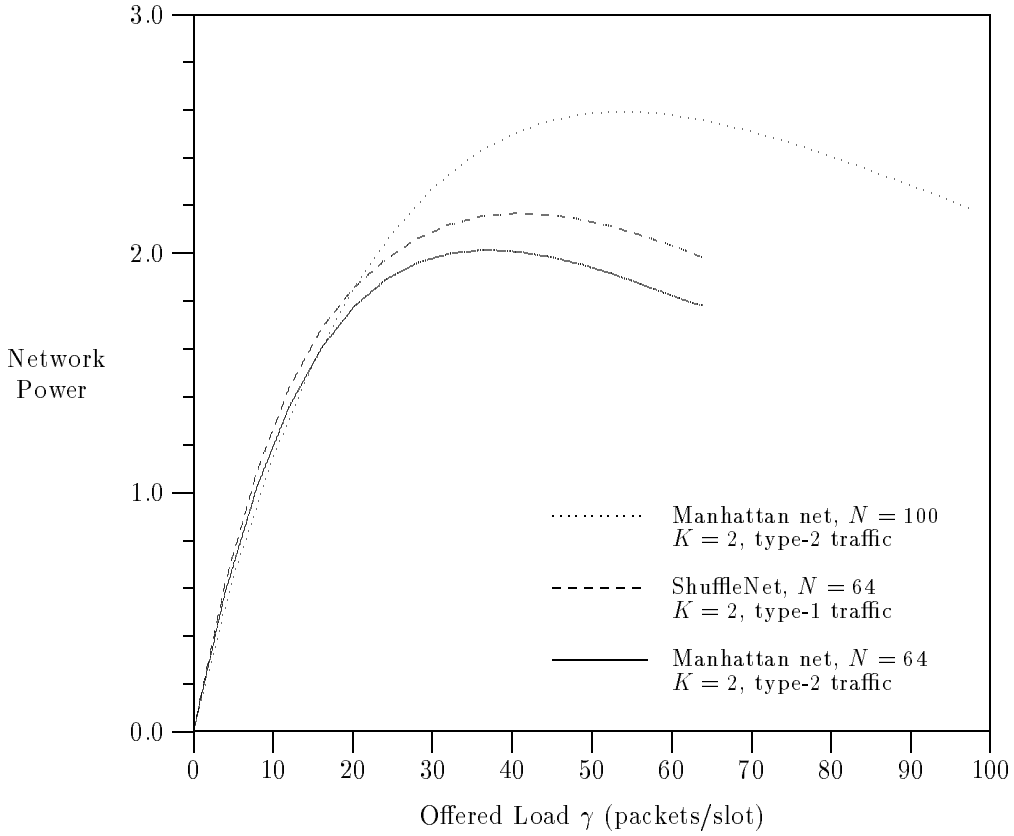


Figure 12: Predicted maximum power curves versus offered load γ in three networks under nonuniform traffic loads

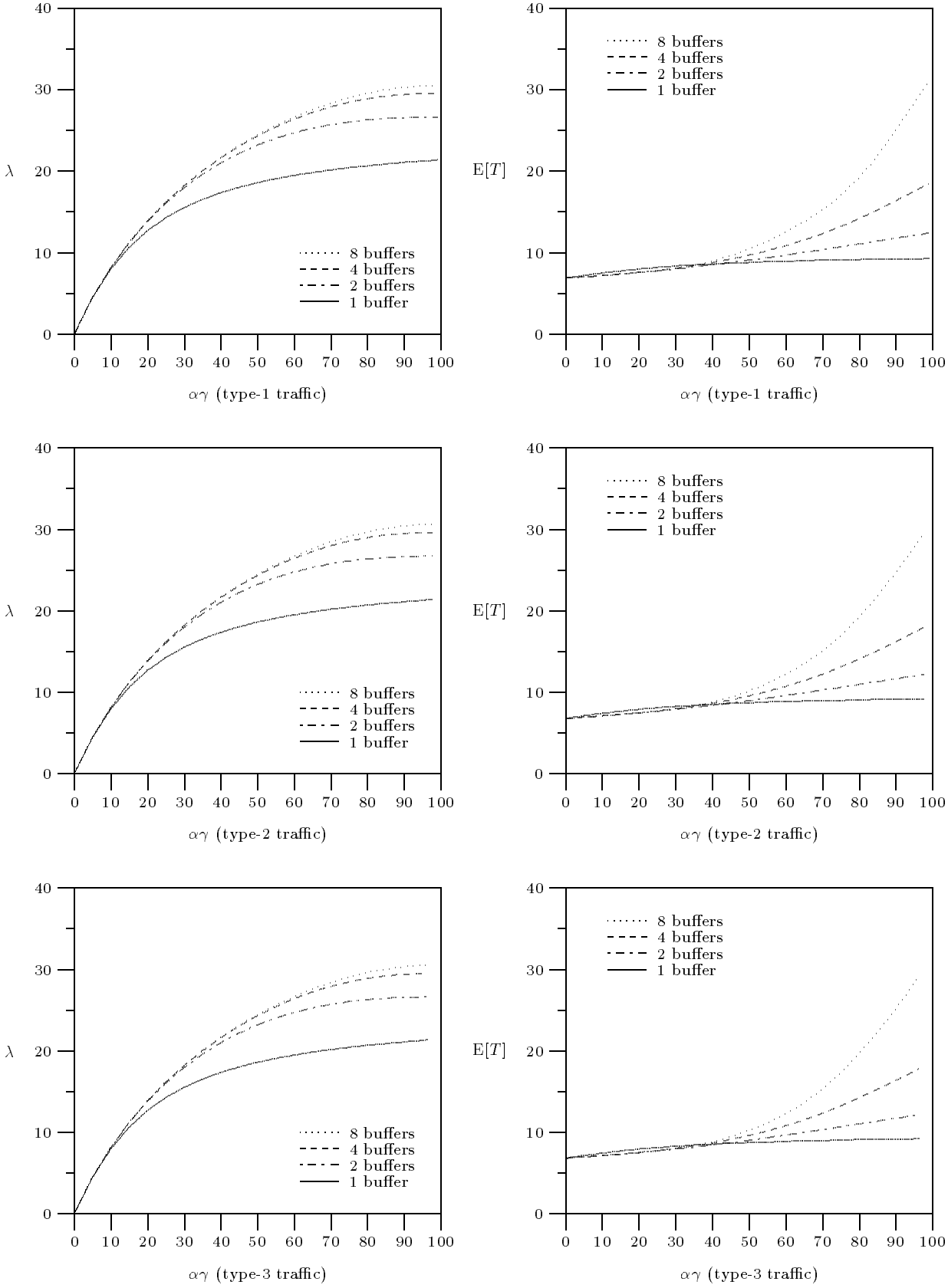


Figure 11: Predicted throughput λ (in packets per time slot) and delay E[T] (in time slots) versus scaled offered load $\alpha\gamma$ (in packets per time slot) in a 100-station multibuffer network with the 10×10 Manhattan street topology under nonuniform traffic loads

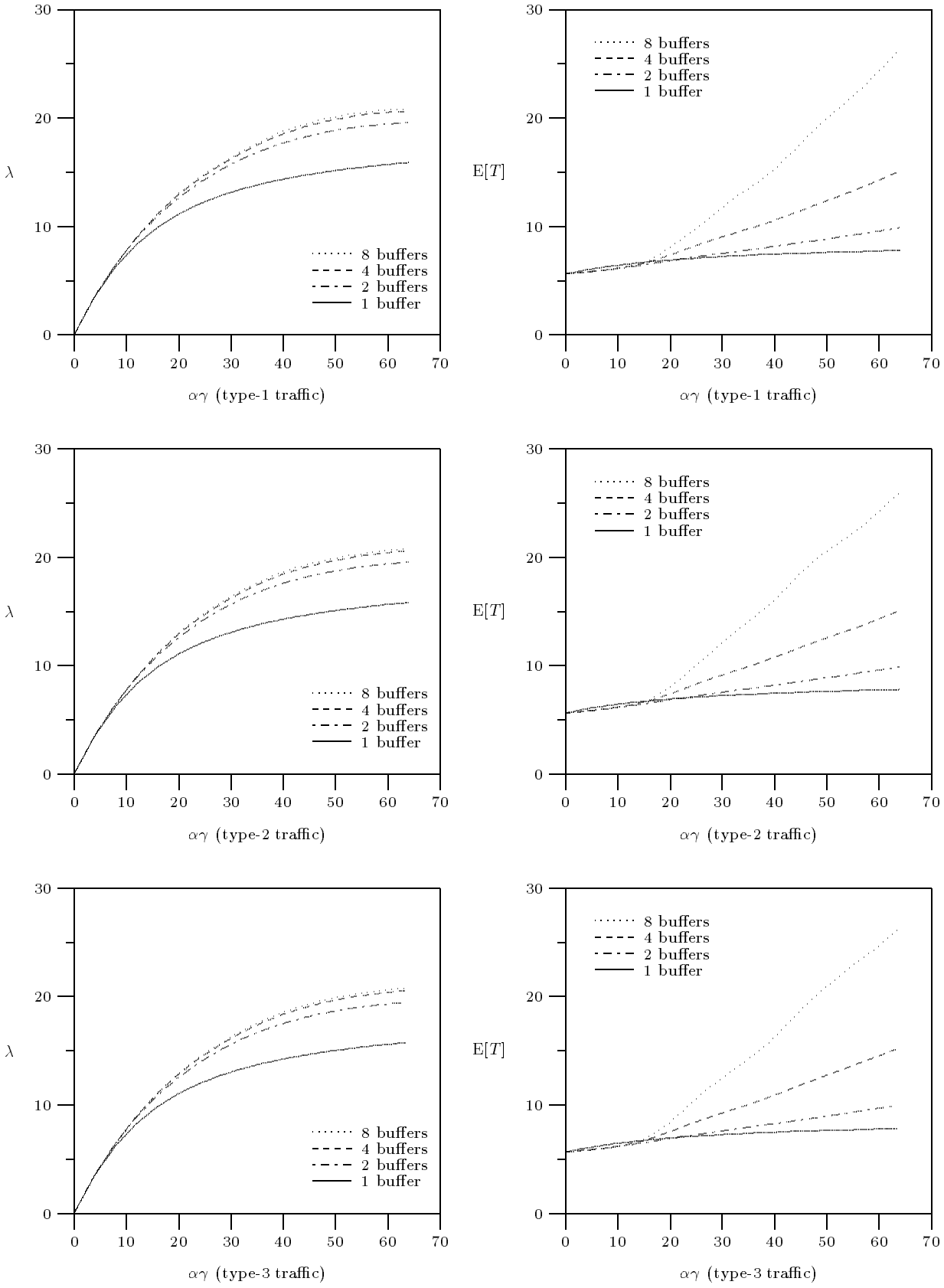


Figure 10: Predicted throughput λ (in packets per time slot) and delay E[T] (in time slots) versus scaled offered load $\alpha\gamma$ (in packets per time slot) in a 64-station multibuffer network with the binary ShuffleNet topology under nonuniform traffic loads

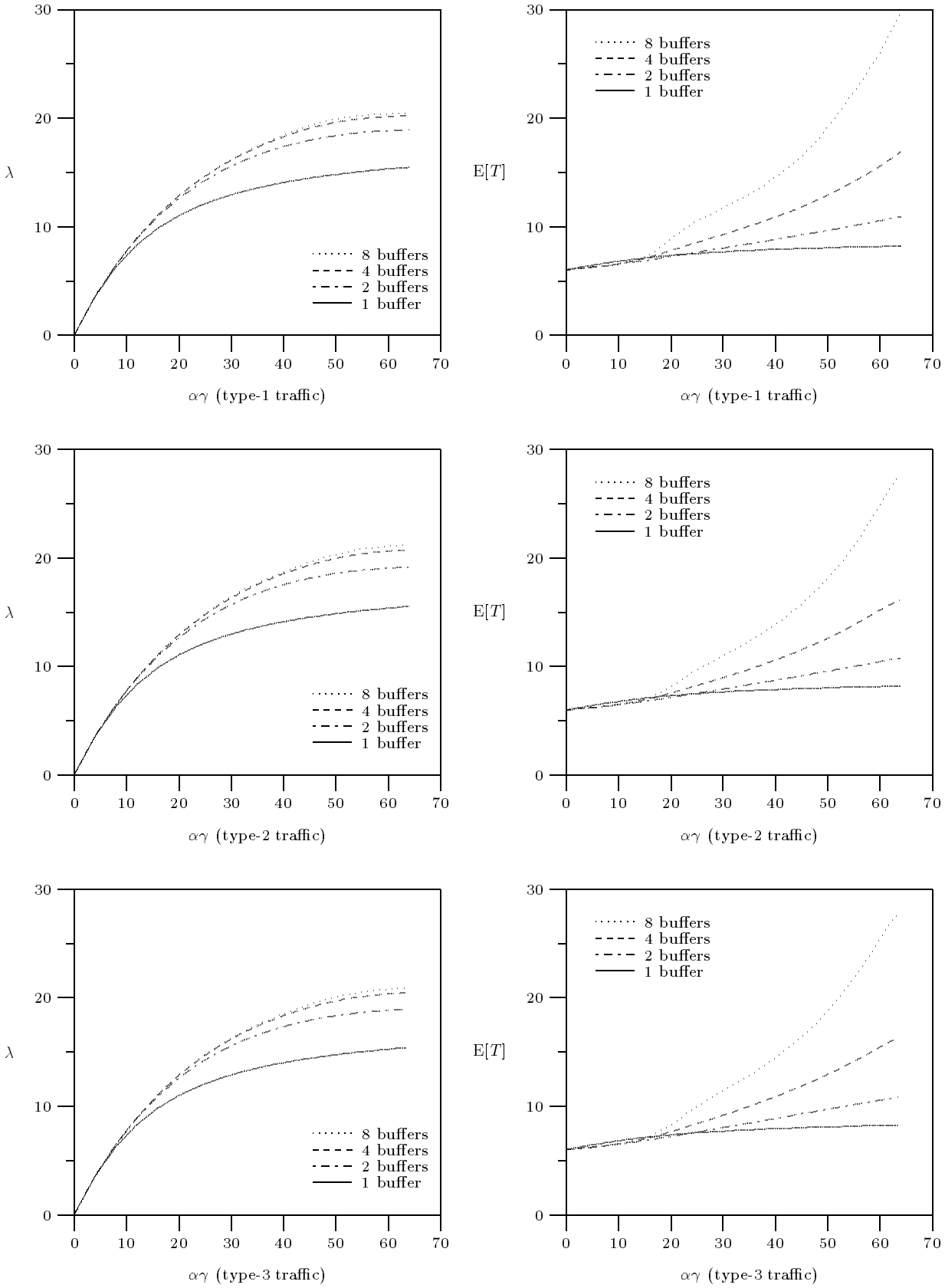


Figure 9: Predicted throughput λ (in packets per time slot) and delay E[T] (in time slots) versus scaled offered load $\alpha\gamma$ (in packets per time slot) in a 64-station multibuffer network with the 8×8 Manhattan street topology under nonuniform traffic loads

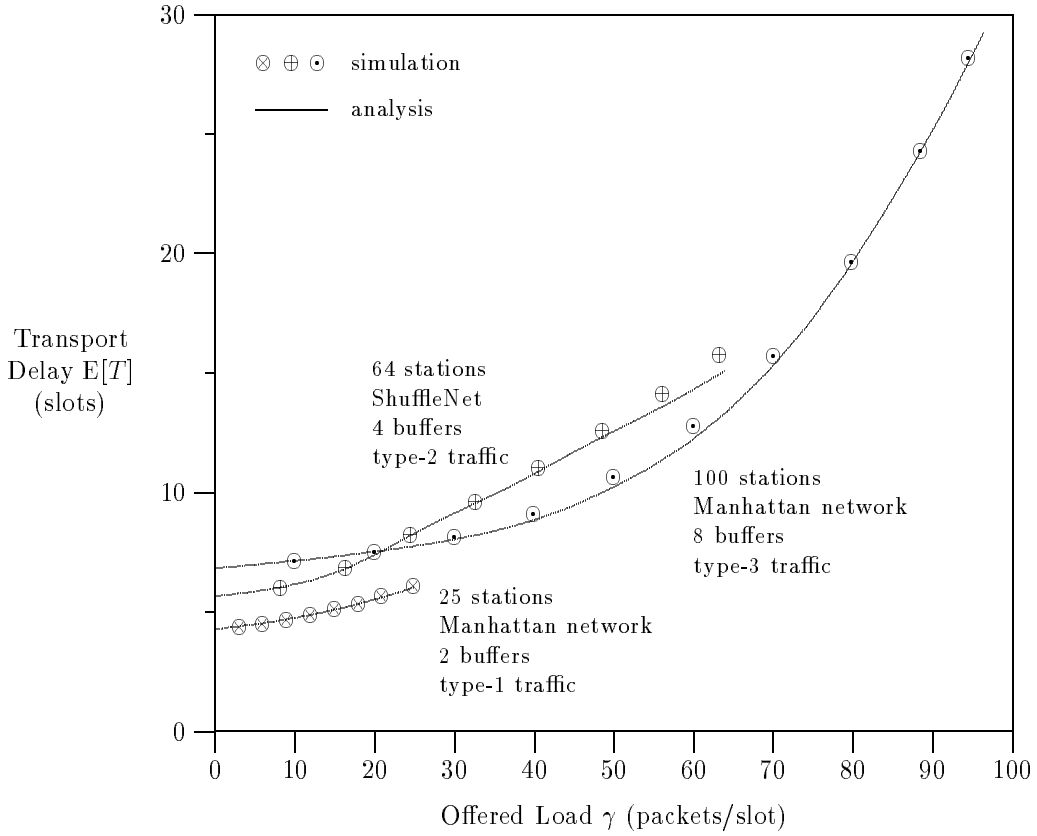


Figure 8: Comparison of analytical and simulation models for three multibuffer networks and nonuniform traffic loads

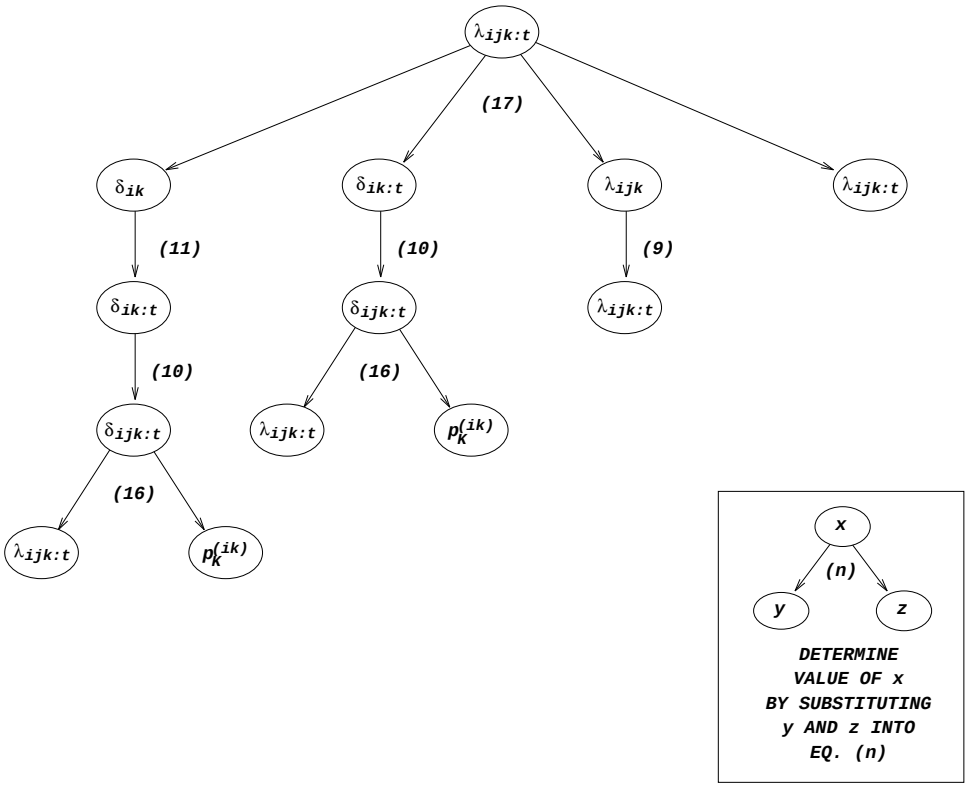


Figure 7: Substitution tree for $\boldsymbol{\lambda} = \boldsymbol{g}(\boldsymbol{p}, \boldsymbol{\lambda})$

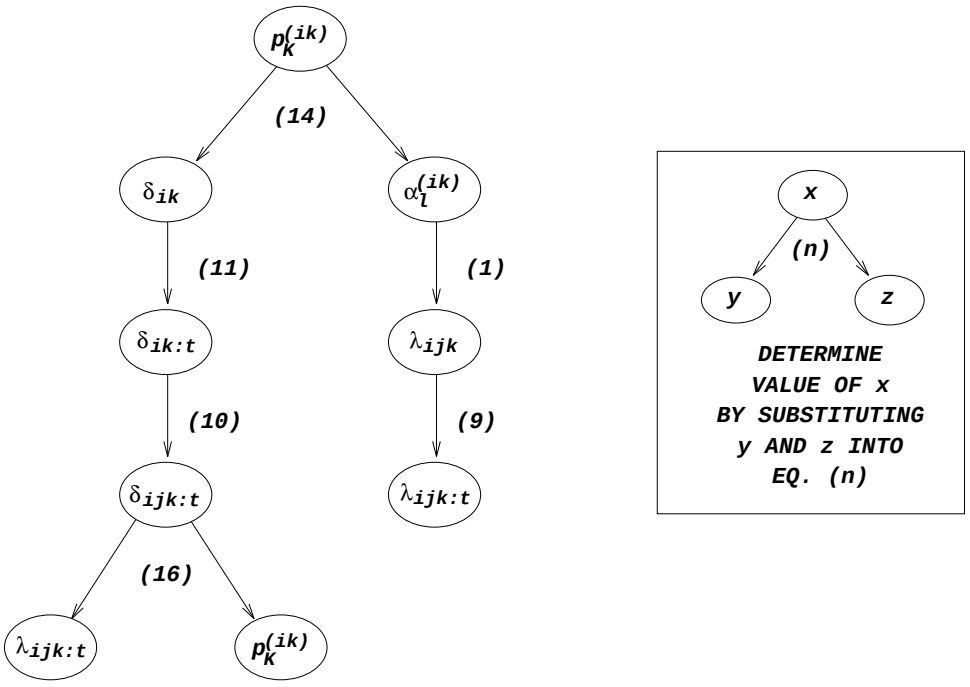


Figure 6: Substitution tree for $\boldsymbol{p} = \boldsymbol{f}(\boldsymbol{p}, \boldsymbol{\lambda})$

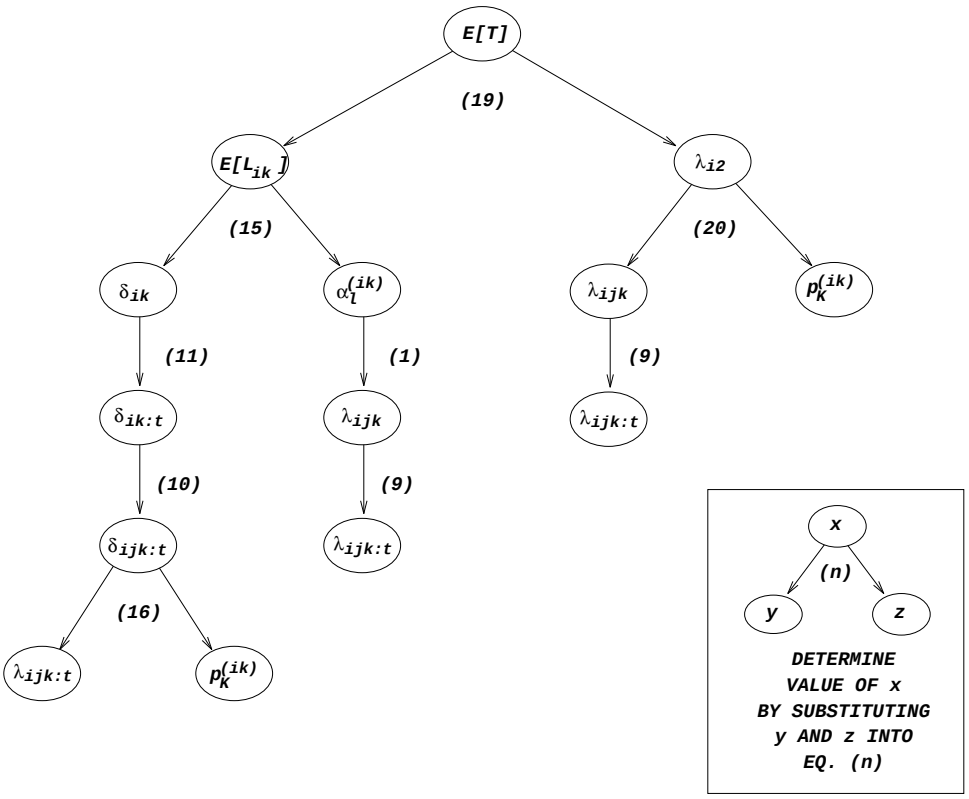


Figure 5: Substitution tree for $\mathrm{E}[T] = F(\boldsymbol{p}, \boldsymbol{\lambda})$

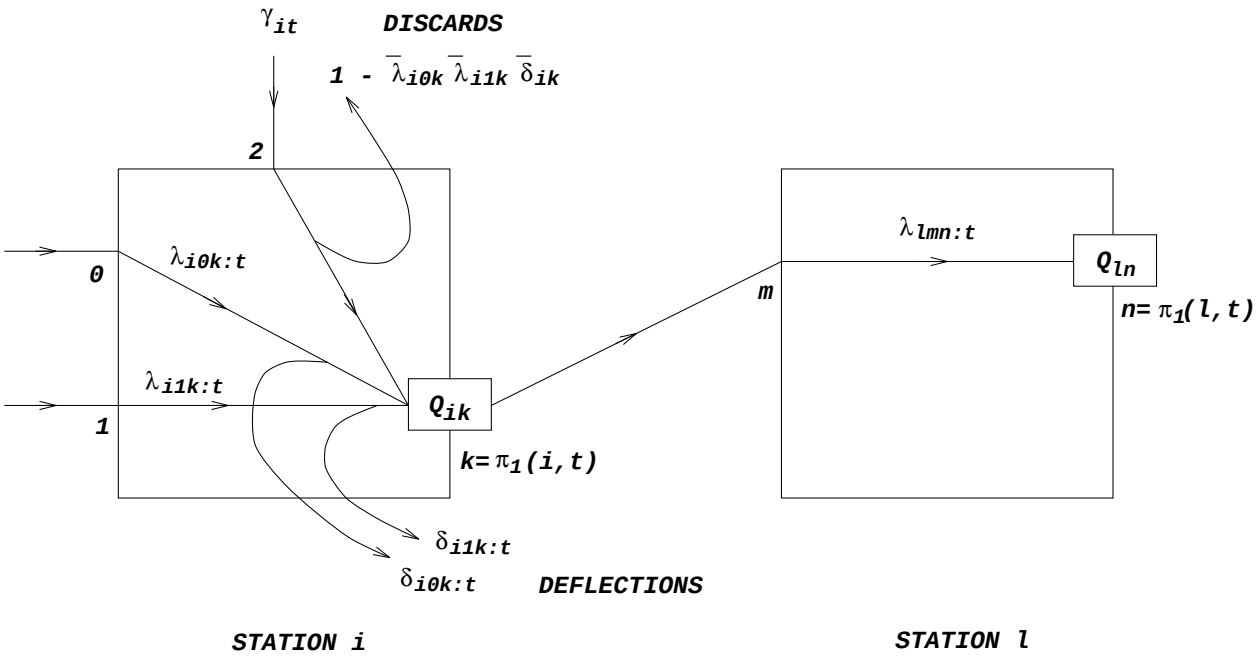


Figure 4: Conservation of t-flow at a queueing center

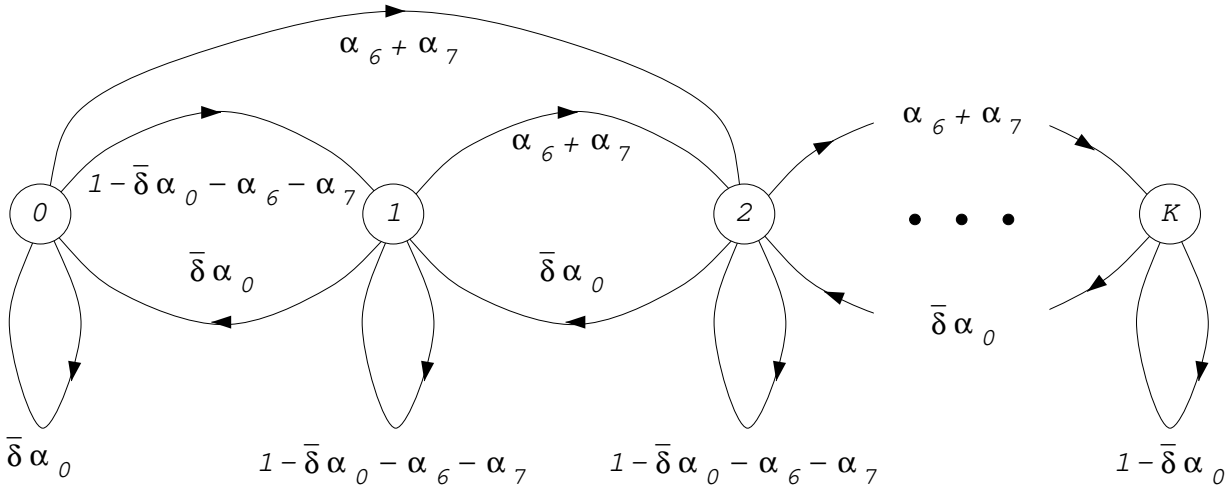
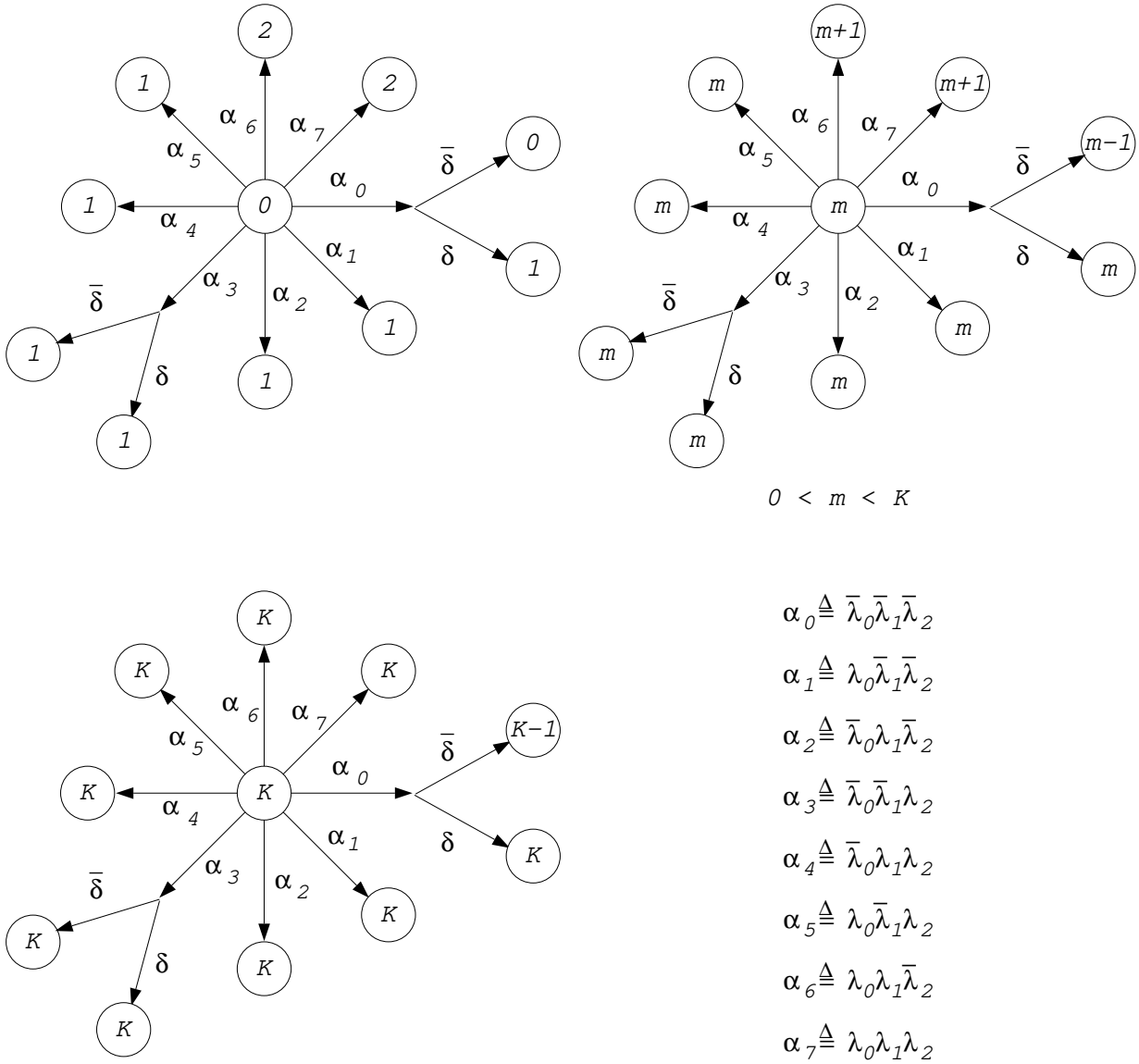


Figure 3: The state-transition graph for the Markov chain



$$\alpha_0 \triangleq \bar{\lambda}_0\bar{\lambda}_1\bar{\lambda}_2$$

$$\alpha_1 \triangleq \lambda_0\bar{\lambda}_1\bar{\lambda}_2$$

$$\alpha_2 \triangleq \bar{\lambda}_0\lambda_1\bar{\lambda}_2$$

$$\alpha_3 \triangleq \bar{\lambda}_0\bar{\lambda}_1\lambda_2$$

$$\alpha_4 \triangleq \bar{\lambda}_0\lambda_1\lambda_2$$

$$\alpha_5 \triangleq \lambda_0\bar{\lambda}_1\lambda_2$$

$$\alpha_6 \triangleq \lambda_0\lambda_1\bar{\lambda}_2$$

$$\alpha_7 \triangleq \lambda_0\lambda_1\lambda_2$$

Figure 2: State transitions arising from different arrival patterns

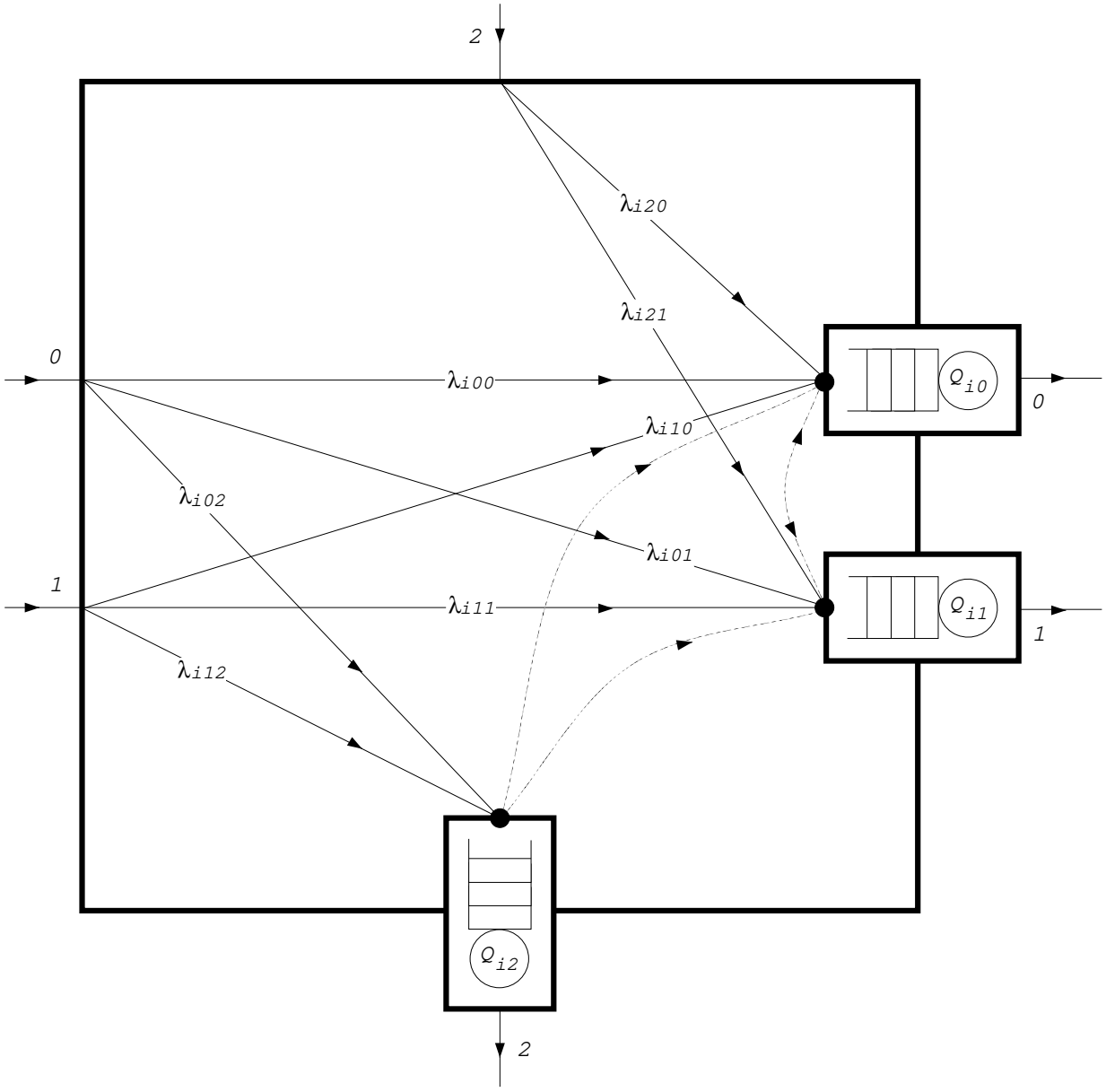


Figure 1: A queueing model of the station

[20] Joseph Bannister, Flaminio Borgonovo, and Mario Gerla. A procedure to evaluate the mean transport time in multibuffer deflection-routing networks with nonuniform traffic. In *Proceedings of IEEE INFOCOM '92*, volume 3, pages 1069–1078, Florence, Italy, May 1992.

[21] Flaminio Borgonovo and Luigi Fratta. Deflection networks: architectures for metropolitan and wide area networks. *Computer Networks and ISDN Systems*, 24(2):171–183, April 1992.

[22] Joseph Bannister, Flaminio Borgonovo, Luigi Fratta, and Mario Gerla. A performance model of deflection routing and its application to the topological design of multichannel networks. Technical Report CSD-910002, UCLA Computer Science Department, Los Angeles, California, January 1991.

[23] Annie Gravey, Jean-Raymond Louvion, and Pierre Boyer. On the Geo/D/1 and Geo/D/1/n queues. *Performance Evaluation*, 11(2):117–125, July 1990.

[24] Leonard Kleinrock. *Queueing Systems, Volume I: Theory.* John Wiley and Sons, New York, New York, 1975.

[25] S. D. Conte and Carl de Boor. *Elementary Numerical Analysis: An Algorithmic Approach.* McGraw–Hill, New York, New York, 1980.

[26] Leonard Kleinrock. Power and deterministic rules of thumb for probabilistic problems in computer communications. In *Proceedings of ICC '79*, pages 43.1.1–43.1.10, Boston, Massachusetts, June 1979.

[27] N. F. Maxemchuk. Comparison of deflection and store-and-forward techniques in the Manhattan street and shuffle-exchange networks. In *Proceedings of IEEE INFOCOM '89*, volume 3, pages 800–809, Ottawa, Canada, April 1989.

[7] Anthony S. Acampora and Syed I. A. Shah. Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing. *IEEE Transactions on Communications*, 40(6):1082–1090, June 1992.

[8] John Thomas Brassil. *Deflection Routing in Certain Regular Networks.* PhD thesis, Electrical Engineering Department, University of California, San Diego, California, 1991.

[9] Abhijit K. Choudhury and Victor O. K. Li. Performance analysis of deflection routing in the Manhattan street network. In *Proceedings of IEEE ICC '91*, pages 1707–1711, Denver, Colorado, June 1991.

[10] Abhijit K. Choudhury and Victor O. K. Li. An approximate analysis of the performance of deflection routing in regular networks. *IEEE Journal on Selected Areas of Communications*, 11(10):1302–1316, October 1993.

[11] Bruce Hajek. Bounds on evacuation time for deflection routing. *Distributed Computing*, 5(1):1–6, 1991.

[12] Arvind Krishna and Bruce Hajek. Performance of shuffle-like switching networks with deflection. In *Proceedings of IEEE INFOCOM '90*, volume 2, pages 473–480, San Francisco, California, June 1990.

[13] Albert G. Greenberg and Bruce Hajek. Deflection routing in hypercube networks. *IEEE Transactions on Communications*, 40(6):1070–1081, June 1992.

[14] Ted H. Szymanski. An analysis of "hot-potato" routing in a fiber optic packet switched hypercube. In *Proceedings of IEEE INFOCOM '90*, volume 3, pages 918–925, San Francisco, California, June 1990.

[15] Chien Fang and Ted H. Szymanski. An analysis of deflection routing in multidimensional regular mesh networks. In *Proceedings of IEEE INFOCOM '91*, volume 2, pages 859–868, Bal Harbour, Florida, April 1991.

[16] Jack Brassil and Rene Cruz. Nonuniform traffic in the Manhattan street network. In *Proceedings of IEEE ICC '91*, pages 1647–1651, Denver, Colorado, June 1991.

[17] Jack Brassil and Rene Cruz. Nonuniform traffic in the Manhattan street network. *Performance Evaluation*, 22, 1995. to appear.

[18] Joseph Bannister, Flaminio Borgonovo, Luigi Fratta, and Mario Gerla. A versatile model for predicting the performance of deflection-routing networks. *Performance Evaluation*, 16(1–3):201–222, November 1992.

[19] D. B. Wagner. Approximate mean-value analysis of deflection-routed shuffle-loop networks. Technical Report CU-CS-608-92, Computer Science Department, University of Colorado, Boulder, Colorado, August 1992.

X	random variable for generating the traffic matrix
$\mathbf{x}$	vector variable
ξ^*	scalar minimizer
$\mathbf{y}$	vector variable
$\|\mathbf{x}\|$	norm of vector $\mathbf{x}$
$\mathbf{x} \cdot \mathbf{y}$	inner product of vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{A}^{\mathrm{T}}$	transpose of matrix $\mathbf{A}$
∇	gradient operator

Acknowledgments

The authors thank Milan Kovačević and Zhensheng Zhang for their critical comments and Mike Meyer for his editorial assistance.

References

[1] Flaminio Borgonovo, Luigi Fratta, and Joseph Bannister. Design of optical deflection-routing networks. In *Proceedings of IEEE INFOCOM '94*, volume 1, pages 120–129, Toronto, Canada, June 1994.

[2] Flaminio Borgonovo, Luigi Fratta, and Joseph Bannister. Unslotted deflection routing in all-optical networks. In *Proceedings of IEEE GLOBECOM '93*, volume 1, pages 119–125, Houston, Texas, November 1993.

[3] Albert G. Greenberg and Jonathan Goodman. Sharp approximate models of adaptive routing in mesh networks. In *Proceedings of the 1986 International Seminar on Teletraffic Analysis and Computer Performance Evaluation*, pages 255–270, Amsterdam, The Netherlands, June 1986.

[4] Terence D. Todd and Allan M. Bignell. Performance modeling of the SIGnet MAN backbone. In *Proceedings of IEEE INFOCOM '90*, volume 1, pages 192–199, San Francisco, California, June 1990.

[5] Flaminio Borgonovo and Enrico Cadorin. Locally-optimal deflection routing in the bidirectional Manhattan network. In *Proceedings of IEEE INFOCOM '90*, volume 2, pages 458–464, San Francisco, California, June 1990.

[6] Zhensheng Zhang and Anthony S. Acampora. Performance analysis of multihop lightwave networks with hot potato routing and distance-age-priorities. In *Proceedings of IEEE INFOCOM '91*, volume 3, pages 1012–1021, Bal Harbour, Florida, April 1991.

ϵ	tolerance value for determining convergence in the algorithm
$\boldsymbol{f}(\cdot)$	vector function of $\boldsymbol{p}$ and $\boldsymbol{\lambda}$
$F(\cdot)$	function for computing E$[T]$
$\boldsymbol{g}(\cdot)$	vector function of $\boldsymbol{p}$ and $\boldsymbol{\lambda}$
$\boldsymbol{G}(\cdot)$	vector function with fixed point $(\boldsymbol{p}, \boldsymbol{\lambda})$
$h(\cdot)$	vector-argument function to be minimized
$h^*(\cdot)$	scalar function to be minimized
$(i,j) \rightarrow (l,m)$	output port j of station i is connected to input port m of station l
K	number of packet buffers in a queueing center
L	number of packets in the isolated queueing center (random variable)
L_{ik}	number of packets in queueing center $\mathcal{Q}_{ik}$ (random variable)
λ	network throughput (in packets per time slot)
$\boldsymbol{\lambda}$	matrix of internal-link flows
λ_j	probability that a class-j packet is generated during a given time slot
λ_{ik}	throughput (in packets per time slot) of queueing center $\mathcal{Q}_{ik}$
λ_{ijk}	probability that there is a packet on internal link (i,j,k) at the start of a given time slot
$\lambda_{ijk:t}$	probability that there is a t-packet on internal link (i,j,k) at the start of a given time slot
$\tilde{\lambda}_{ijk}$	analog of λ_{ijk} in an infinite-buffer network
$\tilde{\lambda}_{ijk:t}$	analog of $\lambda_{ijk:t}$ in an infinite-buffer network
N	number of stations in the network
p_m	probability that m packets occupy the buffers of the isolated queueing center in equilibrium
$p_m^{(n)}$	probability that m packets occupy the buffers of the isolated queueing center during time slot n
$p_m^{(ik)}$	probability that m packets occupy the buffers of queueing center $\mathcal{Q}_{ik}$ in equilibrium
$p_i(t)$	priority of a packet that arrives at station i and is destined for station t
$\tilde{p}_m^{(ik)}$	analog of $p_m^{(ij)}$ in an infinite-buffer network
$\boldsymbol{p}$	matrix of buffer-overflow probabilities
$\mathbf{p}$	equilibrium-state–probability vector of a Markov chain
$\mathbf{P}$	one-step–transition probability matrix of a Markov chain
P$[\cdot]$	probability of an event occurring
$\pi_n(i,t)$	nth-preference output port of a packet that arrives at station i and is destined for station t
$\mathcal{Q}_{ik}$	queueing center associated with the kth output port of station i
$\sum$	summation operator
T	packet transport time (random variable)
x_i	ith entry of vector $\mathbf{x}$
$\bar{x}$	$1 - x$ (complement of x)

Setting $\xi^* = -\mathbf{y} \cdot \nabla h(\mathbf{x})/|\mathbf{A}\mathbf{y}|^2$ and recalling that $\nabla h(\mathbf{x}) = \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{x} - \mathbf{b})$, we substitue ξ^* into Eq. (39) to observe that $\mathrm{d}h^*(\xi)/\mathrm{d}\xi$ vanishes at ξ^*:

$$\begin{aligned}
\left.\frac{\mathrm{d}h^*(\xi)}{\mathrm{d}\xi}\right|_{\xi^*} &= \xi^*|\mathbf{A}\mathbf{y}|^2 + \sum_i (\mathbf{A}_i\mathbf{y})(\mathbf{A}_i\mathbf{x} - b_i) \\
&= -\frac{\mathbf{y}^{\mathrm{T}}\nabla h(\mathbf{x})}{|\mathbf{A}\mathbf{y}|^2}|\mathbf{A}\mathbf{y}|^2 + \sum_i (\mathbf{A}_i\mathbf{y})(\mathbf{A}_i\mathbf{x} - b_i) \\
&= -(\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}})(\mathbf{A}\mathbf{x} - \mathbf{b}) + \sum_i (\mathbf{A}_i\mathbf{y})(\mathbf{A}_i\mathbf{x} - b_i) \\
&= -\sum_i (\mathbf{y}\mathbf{A}_i)(\mathbf{A}_i\mathbf{x} - b_i) + \sum_i (\mathbf{A}_i\mathbf{y})(\mathbf{A}_i\mathbf{x} - b_i) \\
&= 0
\end{aligned}$$

Since $\mathrm{d}h^*(\xi)/\mathrm{d}\xi$ is zero at ξ^* and $\mathrm{d}^2h^*(\xi)/\mathrm{d}\xi^2 \geq 0$, $h^*(\cdot)$ has a minimum at ξ^*.

D Notation

a_{ij}	entry of matrix $\mathbf{A}$
$\mathbf{A}$	matrix used to compute the initial flow vector
$\mathbf{A}_i$	ith row of matrix $\mathbf{A}$
α_j	joint-arrival probability at the isolated queueing center
$\alpha_j^{(ik)}$	joint-arrival probability at queueing center $\mathcal{Q}_{ik}$
$\tilde{\alpha}_j^{(ik)}$	analog of $\alpha_j^{(ik)}$ in an infinite-buffer network
b_i	entry of vector $\mathbf{b}$
$\mathbf{b}$	vector used to compute initial flow vector
β	scaling factor for the traffic matrix
γ	offered traffic load
γ_{st}	probability that the user at station s generates a packet destined for station t during a given time slot
$d(s,t)$	length of the shortest path from station s to station t
$\mathrm{d}y/\mathrm{d}x$	first derivative of y
$\mathrm{d}^2y/\mathrm{d}x^2$	second derivative of y
δ	conditional-arrival probability of a class-3 packet
δ_{ik}	probability that queueing center $\mathcal{Q}_{ik}$ receives a deflected packet during a given time slot
$\delta_{ik:t}$	probability that queueing center $\mathcal{Q}_{ik}$ receives a deflected t-packet during a given time slot
$\delta_{ijk:t}$	probability of deflecting a t-packet away from internal link (i,j,k) during a given time slot
$\Delta\beta$	incremental scaling factor for the traffic matrix
$\partial y/\partial x$	partial derivative of y
$\mathrm{E}[\cdot]$	expectation operator

Define $(c_1, c_2, \ldots, c_n)^{\mathrm{T}} \triangleq \mathbf{A}\mathbf{x} - \mathbf{b}$, i.e.,

$$c_i \triangleq \sum_j a_{ij} x_j - b_i$$

and $(a'_{ki}) = \mathbf{A}^{\mathrm{T}}$, i.e., $a'_{ki} \triangleq a_{ik}$. We now rewrite Eq. (38) as

$$\begin{aligned} \nabla h(\mathbf{x}) &= \left[\sum_i a'_{ki} \left(\sum_j a_{ij} x_j - b_i \right) \right]_{k=1}^{n} \\ &= \left[\sum_i a'_{ki} c_i \right]_{k=1}^{n} \\ &= \mathbf{A}^{\mathrm{T}} (\mathbf{A}\mathbf{x} - \mathbf{b}) \end{aligned}$$

This is Eq. (29).

C Derivation of an expression for the minimizing scalar

We show in this appendix that Eq. (30) determines the value of ξ_n^* required in Eq. (28) of Section 5.

Given the n-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$, we define the following single-variable function:

$$h^*(\xi) \triangleq h(\mathbf{x} + \xi \mathbf{y})$$

where $h(\cdot)$ is a real-valued function of n-dimensional vectors defined by

$$h(\mathbf{x}) = \frac{1}{2} |\mathbf{A}\mathbf{x} - \mathbf{b}|^2$$

Let $\mathbf{A}_i$ be the ith row of $\mathbf{A}$. We start by computing the first two derivatives of $h^*(\cdot)$:

$$\begin{aligned} \frac{\mathrm{d}h^*(\xi)}{\mathrm{d}\xi} &= \frac{\mathrm{d}}{\mathrm{d}\xi} \frac{1}{2} |\mathbf{A}(\mathbf{x} + \xi \mathbf{y}) - \mathbf{b}|^2 \\ &= \frac{\mathrm{d}}{\mathrm{d}\xi} \frac{1}{2} \left| \begin{array}{c} \mathbf{A}_1 \mathbf{x} + \xi \mathbf{A}_1 \mathbf{y} - b_1 \\ \cdots \\ \mathbf{A}_n \mathbf{x} + \xi \mathbf{A}_n \mathbf{y} - b_n \end{array} \right|^2 \\ &= \frac{\mathrm{d}}{\mathrm{d}\xi} \frac{1}{2} \sum_i (\xi \mathbf{A}_i \mathbf{y} + \mathbf{A}_i \mathbf{x} - b_i)^2 \\ &= \frac{\mathrm{d}}{\mathrm{d}\xi} \frac{1}{2} \sum_i \xi^2 (\mathbf{A}_i \mathbf{y})^2 + 2\xi (\mathbf{A}_i \mathbf{y})(\mathbf{A}_i \mathbf{x} - b_i) + (\mathbf{A}_i \mathbf{x} - b_i)^2 \\ &= \xi |\mathbf{A}\mathbf{y}|^2 + \sum_i (\mathbf{A}_i \mathbf{y})(\mathbf{A}_i \mathbf{x} - b_i) \end{aligned} \tag{39}$$

Therefore

$$\frac{\mathrm{d}^2 h^*(\xi)}{\mathrm{d}\xi^2} = |\mathbf{A}\mathbf{y}|^2$$

and the second derivative is everywhere nonnegative.

From Eq. (31) it is obvious that

$$p_1 = p_0 \frac{1 - \bar{\delta}\alpha_0}{\bar{\delta}\alpha_0} \tag{35}$$

Since Eq. (32) expresses p_2 in terms of p_1 and p_0, we apply Eq. (35) to obtain

$$p_2 = p_0 \frac{\alpha_6 + \alpha_7}{(\bar{\delta}\alpha_0)^2} \tag{36}$$

Similarly, from Eqs. (33), (35), and (36) we obtain

$$p_3 = p_0 \frac{(\alpha_6 + \alpha_7)^2}{(\bar{\delta}\alpha_0)^3}$$

Continuing this line of approach, we see from Eq. (34) that the general expression for p_m $(2 \leq m \leq K)$ is

$$p_m = p_0 \frac{(\alpha_6 + \alpha_7)^{m-1}}{(\bar{\delta}\alpha_0)^m} \tag{37}$$

Substituting Eqs. (35) and (37) into Eq. (3), we obtain the following expression involving p_0:

$$1 = p_0 + p_0 \frac{1 - \bar{\delta}\alpha_0}{\bar{\delta}\alpha_0} + \sum_{m=2}^{K} p_0 \frac{(\alpha_6 + \alpha_7)^{m-1}}{(\bar{\delta}\alpha_0)^m} = p_0 \left[1 + \frac{1 - \bar{\delta}\alpha_0}{\bar{\delta}\alpha_0} + \sum_{m=2}^{K} \frac{(\alpha_6 + \alpha_7)^{m-1}}{(\bar{\delta}\alpha_0)^m} \right]$$

This reduces to

$$p_0 = \frac{(\bar{\delta}\alpha_0)^K (\bar{\delta}\alpha_0 - \alpha_6 - \alpha_7)}{(\bar{\delta}\alpha_0)^K - (\alpha_6 + \alpha_7)^K}$$

which is the same as Eq. (4). Also, Eqs. (35) and (37) reduce to

$$p_1 = \frac{(1 - \bar{\delta}\alpha_0)(\bar{\delta}\alpha_0 - \alpha_6 - \alpha_7)(\bar{\delta}\alpha_0)^{K-1}}{(\bar{\delta}\alpha_0)^K - (\alpha_6 + \alpha_7)^K}$$

and for $2 \leq m \leq K$

$$p_m = \frac{(\bar{\delta}\alpha_0 - \alpha_6 - \alpha_7)(\bar{\delta}\alpha_0)^{K-m}(\alpha_6 + \alpha_7)^{m-1}}{(\bar{\delta}\alpha_0)^K - (\alpha_6 + \alpha_7)^K}$$

which are the same as Eqs. (5) and (6).

B Derivation of the expression for the gradient vector

We derive here the expression—given by Eq. (29) of Section 5—for the gradient of $h(\cdot)$.

Let $\mathbf{x} \triangleq (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$, $\mathbf{b} \triangleq (b_1, b_2, \ldots, b_n)^{\mathrm{T}}$, and $\mathbf{A}_i \triangleq (a_{ij})$. Recalling that $h(\mathbf{x}) = |\mathbf{A}\mathbf{x} - \mathbf{b}|^2/2$, we compute the gradient vector of $h(\cdot)$ as follows:

$$\begin{aligned}
\nabla h(\mathbf{x}) &= \left[\frac{\partial h(\mathbf{x})}{\partial x_k} \right]_{k=1}^{n} \\
&= \left[\frac{\partial}{\partial x_k} \frac{1}{2} \sum_i \left(\sum_j a_{ij} x_j - b_i \right)^2 \right]_{k=1}^{n} \\
&= \left[a_{ik} \left(\sum_i \sum_j a_{ij} x_j - b_i \right) \right]_{k=1}^{n}
\end{aligned} \tag{38}$$

ticular, we have modeled a network with no user input buffers, a specific admission-control policy, and destination-based priorities. Each of these choices has other alternatives. The absence of user input buffers adversely affects the performance of the network, since blocked packets must be discarded. A more realistic model might allow for the use of a finite number of packet buffers at the user input port. The analysis of such a model could be accomplished using the techniques of [18]. As have other researchers, we give in-transit packets absolute priority over user-generated packets, but this is by no means necessary or beneficial. Our model could be modified easily to accommodate an admission-control policy that allows user-generated packets to be queued (rather than discarded) if there is an unoccupied packet buffer, but the cost in terms of modeling is that the underlying Markov process of the queueing center would no longer resemble a birth–death process. Thus, the analysis of such an admission-control policy would be more complex. There is also a question of whether such a liberalized admission policy would control congestion as well as the one studied here. We are currently studying these problems and their solutions.

We were originally motivated to develop analytical models of deflection routing while studying the problem of optimizing the topology of deflection-routing networks [22]. This problem requires us to evaluate the performance of a large number of topologies and, therefore, demands an efficient technique for evaluating the performance of each topology. Although the process is demonstrably faster than simulation, problems such as topology optimization, which is solved by means of simulated annealing, need evaluation algorithms that are even faster than the proposed algorithm. The development of fast algorithms that can be used to evaluate the performance of deflection-routing networks is thus essential to solving the problem of topology optimization. For example, by using a better estimate of the initial link-flow rates, one could improve the algorithm's speed in the case of heavy traffic load.

Appendix

A Markov-chain solution

We show here that Eqs. (4) through (6) specify the equilibrium-state probability distribution for the discrete-time Markov chain defined in Section 3.

To solve Eqs. (2) and (3), we solve the balance equations for the state diagram of Fig. 3. Equating the outward and inward probability flux at each state, we write the following equations:

$$p_0(1-\bar{\delta}\alpha_0) = p_1\bar{\delta}\alpha_0 \tag{31}$$

$$p_1(\bar{\delta}\alpha_0+\alpha_6+\alpha_7) = p_0(1-\bar{\delta}\alpha_0-\alpha_6-\alpha_7)+p_2\bar{\delta}\alpha_0 \tag{32}$$

$$p_2(\bar{\delta}\alpha_0+\alpha_6+\alpha_7) = p_0(\alpha_6+\alpha_7)+p_1(\alpha_6+\alpha_7)+p_3\bar{\delta}\alpha_0 \tag{33}$$

$$p_m(\bar{\delta}\alpha_0+\alpha_6+\alpha_7) = p_{m-1}(\alpha_6+\alpha_7)+p_{m+1}\bar{\delta}\alpha_0 \tag{34}$$

for $3 \le m < K$.

desired output port. In a busy network, therefore, relatively few user-generated packets can enter the network. The maximum throughput falls short of the theoretical limit of N packets per time slot by a wide margin, e.g. the networks of Figs. 9 and 10 achieve a maximum throughput of about 20 packets per time slot, which is only 30 percent of the theoretical limit of 64 packets per time slot.

It is interesting to compare the 64-station ShuffleNet to the 64-station Manhattan street network. ShuffleNet generally outperforms the Manhattan street network in the areas of throughput, delay, and power. The Manhattan street network, however, excels in specific cases. For example, ShuffleNet's throughput is higher than the Manhattan network's for all cases except when the traffic is of type 2 and there are four or eight buffers, or when the traffic is of type 3 and there are eight buffers. Similarly, ShuffleNet's delay is less than the Manhattan network's except when there are eight buffers; in that case, neither network's delay curve dominates the other's over the entire range of offered load. Especially when larger buffers are provided, the delay curve of ShuffleNet has a much sharper knee in its transition from the low-load to the high-load region, whereas the Manhattan network's delay curve has a gentler rise. The nearly linear growth in ShuffleNet's delay results in ShuffleNet achieving lower delay than the Manhattan network when operating at full capacity.

In a given network traffic types 1 through 3 produce similar performance characteristics, as may be seen from Figs. 9 through 11. To determine the extent to which different traffic patterns can affect performance in a network, we consider a variant of type-3 traffic, in which a source station sends a fixed amount of traffic to a single destination station. The traffic pattern is described by a permutation that maps each source station to a unique destination station. We use a simple rotation to map station i to station $i+m$ (modulo N), where m is a fixed offset and $0 \leq i < N$. We show in Fig. 13 the throughput and mean transport delay vs. offered load in a 5×5 Manhattan network with one buffer per output port. The four traffic matrices correspond to rotations by offsets of 1, 9, 23, and 24. The curves span a range of performance levels, differing by as much as 50 percent. We thus see that traffic patterns can exert a significant effect on performance.

7 Conclusion

The use of independence assumptions has allowed us to derive an approximate model of deflection-routing networks that have multibuffer stations and two-connected topologies. We have presented a procedure for estimating the mean transport time (and throughput) of these networks, even when the offered traffic is nonuniform. To date, no known multibuffer deflection-routing models have been applicable to nonuniform traffic, and evaluations had to be done by means of computationally intensive simulations. The proposed procedure is efficient and yields very accurate results, as our comparisons with simulations have confirmed. The model is also quite general and can be applied to deflection-routing networks that deflect packets according to destination-based priorities.

We have modeled only a few of the several options for designing a deflection-routing network. In par-

per station, respectively. Two topologies are studied: the Manhattan street network, which is based on the two-dimensional torus, and ShuffleNet, which is based on the recirculating perfect shuffle. The 25-station network is loaded with type-1 traffic, the 64-station network with type-2 traffic, and the 100-station network with type-3 traffic. The simulation results are the averages of 10 independent runs, each of which spans 3000 time slots. The average values are tightly bracketed by 95-percent confidence intervals, which are not shown in the graphs because they are virtually indistinguishable from the plotted averages. The agreement between the simulation and analytical models is close, with the analytical model generally overestimating slightly the actual transport delay.

Figs. 9 through 11 plot the analytically obtained throughput and transport delay of a 64-station Manhattan network, a 64-station ShuffleNet, and a 100-station Manhattan network. The buffer sizes modeled are one, two, four, and eight, and all three traffic types are examined. We see little variation in performance as the traffic matrix changes. In both Manhattan street networks the performance is highest under type-2 traffic, while in the ShuffleNet performance is highest under type-1 traffic. The graphs for these three networks demonstrate clearly that the addition of but one additional buffer increases throughput significantly while maintaining low delay. In both the 64- and 100-station networks, two buffers provide a maximum throughput that is about 90 percent of the maximum throughput provided by eight buffers. However, the transport delay achieved with two buffers is well below half that achieved with eight buffers, which indicates that at high loads the queueing time—rather than the increase in path length that results from deflections—makes a dominant contribution to the overall delay. Moreover, at low loads the delay of the two-buffer network is lower than even the single-buffer network. This is presumably because of the path-length growth that results from deflections in the single-buffer network, whereas more buffers at low loads protect the network from excessive deflections. In fact, for a given traffic type and topology, the two-buffer network achieves higher power—defined as the ratio of throughput to delay [26]—than any other buffer size. For each of the three networks studied, we show in Fig. 12 the graph of the power curve with the highest peak value. The fact that the use of a small number of buffers will guarantee the highest network power is especially fortunate for all-optical networks, in which deflection routing is the dominant routing technique, since it is very costly to implement an all-optical switch with more than one holding buffer per link [1].

The throughput curves of Figs. 9 through 11 indicate the absence of congestion in the networks. The throughput increases monotonically with the offered load. Simulation and analysis results for other networks and traffic loads also confirm the absence of congestion, even in extreme cases such as destination saturation, in which all stations send packets to a single sink station. These findings are to be contrasted with those of other studies (e.g. [27] and [10]), in which multibuffer Manhattan networks and ShuffleNets are observed to experience differing degrees of congestion, whereby the throughput actually decreases when the offered load exceeds a threshold. The reason for the absence of congestion in Figs. 9 through 11 is the conservative admission-control policy used. The throughput—which has an upper limit of N packets per time slot—remains low because user-generated packets are discarded whenever an in-transit packet is present at the

1. Type-1 traffic: a nonuniform traffic matrix obtained by taking X to be uniformly distributed on the interval $(0, 1)$, i.e.,

$$\mathrm{P}[X \leq x] = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } 0 < x < 1 \\ 1 & \text{if } x \geq 1 \end{cases}$$

2. Type-2 traffic: a nonuniform traffic matrix obtained by taking X to be exponentially distributed with parameter 1, i.e.,

$$\mathrm{P}[X \leq x] = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 - e^{-x} & \text{if } x > 0 \end{cases}$$

3. Type-3 traffic: a nonuniform traffic matrix obtained by taking X to be a Bernoulli random variable with discrete values 0 and a for $a > 2$, i.e.,

$$\mathrm{P}[X = k] = \begin{cases} 1/a & \text{if } k = a \\ 1 - 1/a & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$$

Type-1 traffic models a network in which a station sends unequal—but roughly similar—amounts of traffic to each station. Type-2 traffic models a network in which a station sends varying amounts of traffic to each station. Type-3 traffic models a network in which a station sends varying amounts of traffic to a subset of stations. Note that the mean value of each nondiagonal entry of a traffic matrix thus generated is 1 for each traffic type. The second moment of X influences the degree of variability in the traffic matrix; the higher the value of $\mathrm{E}[X^2]$ the more variable is the traffic matrix (γ_{ij}). The second moments of X for traffic types 1, 2, and 3 are 1/3, 2, and a, respectively, and thus there is increasing variability in the traffic matrices as the traffic type increases. For the remainder of the paper, we restrict ourselves to type-3 traffic matrices with $a = 3$, i.e., each station sends packets to about 1/3 of the other stations.

Given a specific offered traffic matrix (γ_{ij}) of type 1, 2, or 3, we produce a range of scaled matrices $(\alpha\gamma_{ij})$ by choosing different values of α. Obviously, the total load of the scaled-up traffic matrix is $\alpha\gamma$, where $\gamma \triangleq \sum_i \sum_j \gamma_{ij}$. We are interested in how the mean transport time $\mathrm{E}[T]$ and the network throughput λ vary as the traffic load γ is scaled up. Starting with a seed traffic matrix (γ_{ij}) with small γ, we systematically increase α until some row of the scaled matrix sums to 1, i.e., $\sum_j \alpha\gamma_{ij} = 1$ for some i. If row i sums to 1, then station i is at saturation, issuing one packet every time slot. We continue scaling the matrix, leaving alone the entries of those rows that sum to 1. Eventually all rows sum to 1, and the value of $\alpha\gamma$ achieves the maximum of N packets per slot. The three types of traffic that we have chosen have comparable scaling properties, in that their maximum total offered traffic loads are nearly equal.

We show in Fig. 8 a plot of the mean transport delay $\mathrm{E}[T]$ versus the scaled offered load $\alpha\gamma$. The graph compares the values predicted by the analytical model to the simulation results for three different networks. The three networks have 25, 64, and 100 stations, and these networks have two, four, and eight buffers

We again gain efficiency by eliminating the need to perform a systematic search (e.g. a Fibonacci search for a bracketed minimum). Since the computation of $\nabla h(\cdot)$ makes use of only sparse-matrix multiplications, the above expression for ξ_n^* can be calculated rapidly. At each step of the iteration in Eq. (28), we are required to perform only three sparse-matrix multiplications and one inner-product operation.

However, under very high offered loads (i.e., when $\sum_i \sum_j \gamma_{ij}$ is high) the approximation in Eq. (26) is not accurate, and the algorithm must be modified slightly. In such cases there might not be a feasible initial solution to Eq. (26), i.e., we can obtain a solution with $\lambda_{ijk} < 0$ or $\lambda_{ijk} > 1$. Such link flows are not physically possible, because they are not valid probabilities. To overcome this situation we first solve Eq. (26) with a scaled-down offered load by taking the new traffic matrix to be $(\beta\gamma_{st})$ for a choice of β $(0 < \beta < 1)$ that yields a feasible solution. Having found a β for which Eq. (26) has a feasible solution, we use this solution as the starting point of the algorithm by assigning it to $\boldsymbol{\lambda}_0$ in step 1. The algorithm then proceeds until a final value $\boldsymbol{\lambda}_n$ is obtained according to the test of step 5. This final value $\boldsymbol{\lambda}_n$ is a close approximation to the solution when the traffic matrix is scaled up by a small amount: we use $\boldsymbol{\lambda}_n$ as the starting point for the algorithm to solve the link flows in a network with traffic matrix $[(\beta+\Delta\beta)\gamma_{st}]$, where $\Delta\beta$ is a small positive increment. Repeating this process, we eventually obtain the solution for the original, unscaled traffic matrix (γ_{st}).

6 Modeling results

In this section we validate the fidelity of our model by comparing the mean transport delay and throughput predicted by the algorithm with the values measured in a detailed, discrete-event simulation of several deflection-routing networks under various traffic loads. We also use our analytical model to conduct a study of how multibuffer deflection-routing networks perform under nonuniform traffic loads. We consider in our analysis networks of sizes 25, 64, and 100 stations, including both Manhattan street and ShuffleNet topologies.

We will plot values of the mean transport time and network throughput versus the total offered load for three different nonuniform scaled traffic matrices. We specify the traffic matrix by viewing its entries as the values of independent random variables. Thus, we let entry γ_{st} of the traffic matrix be chosen from a probability distribution given by the nonnegative random variable X:

$$\gamma_{st} = \begin{cases} X & \text{if } s \neq t \\ 0 & \text{if } s = t \end{cases}$$

By sampling the random variable X for every nondiagonal entry of (γ_{st}), we can determine a specific nonuniform traffic matrix that has a given amount of "skew" in it. Obviously, the case in which X is a constant random variable corresponds to a uniform traffic matrix, i.e., one in which all nondiagonal entries are identical. We select three different types of traffic matrices in the sequel:

where $\mathbf{A}$ is an $N^2 \times N^2$ matrix, and $\mathbf{x}$ and $\mathbf{b}$ are N^2-dimensional vectors. Now we define the function

$$h(\mathbf{x}) \triangleq \frac{1}{2} |\mathbf{A}\mathbf{x} - \mathbf{b}|^2$$

Clearly, the value of $\mathbf{x}$ that minimizes $h(\cdot)$ also satisfies Eq. (27). We can therefore use an unconstrained optimization procedure to solve Eq. (27).

We use the well-known method of steepest descent [25] to minimize $h(\cdot)$. Given an initial vector $\mathbf{x}_0$, the sequence of vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots$ defined by the iterative relation

$$\mathbf{x}_{n+1} \triangleq \mathbf{x}_n + \xi_n^* \nabla h(\mathbf{x}_n) \tag{28}$$

converges to a vector $\mathbf{x}^*$ such that $h(\mathbf{x}^*)$ is a minimum of $h(\cdot)$; the scalar ξ_n^* is the value of ξ that minimizes the expression $h[\mathbf{x}_n + \xi \nabla h(\mathbf{x}_n)]$. Since $h(\cdot)$ is a convex function, this value is the global minimum, which also satisfies Eq. (27). The iterative procedure defined by Eq. (28) is terminated when the difference between successive values of the function falls below a predetermined tolerance.

We can exploit the sparseness of $\mathbf{A}$ to minimize $h(\mathbf{x})$ efficiently. First, we show in Appendix B that

$$\nabla h(\mathbf{x}) = \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{x} - \mathbf{b}) \tag{29}$$

This equation refers to matrix multiplication by $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$. Specifically, these $N^2 \times N^2$ matrices, whose entries are indexed as pairs of tuples, can be expressed as follows:

$$\mathbf{A}[(i,t),(l,s)] = \begin{cases} 1 & \text{if } \exists m \ [i, \pi_1(i,t)] \rightarrow (l,m) \text{ and } s = t \\ \gamma_{it} - 1 & \text{if } l = i \text{ and } s = t \\ \gamma_{it} & \text{if } l = i, \ \pi_1(l,s) = \pi_1(i,t), \text{ and } s \neq t \\ 0 & \text{otherwise} \end{cases}$$

and

$$\mathbf{A}^{\mathrm{T}}[(l,s),(i,t)] = \begin{cases} 1 & \text{if } \exists m \ [i, \pi_1(i,t)] \rightarrow (l,m) \text{ and } t = s \\ \gamma_{ls} - 1 & \text{if } i = l \text{ and } t = s \\ \gamma_{ls} & \text{if } i = l, \ \pi_1(l,s) = \pi_1(i,t), \text{ and } t \neq s \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq N$, $1 \leq t \leq N$, $1 \leq l \leq N$, and $1 \leq s \leq N$. The sparse-matrix multiplications of Eq. (29) consume far fewer than the usual number of multiplications involved in dense-matrix multiplications; in this case the number is reduced by a factor of about N. Therefore, each step of the iterative procedure of Eq. (28) can be performed efficiently. Compared to solving Eq. (27) by a conventional linear-algebraic technique, such as Gaussian elimination, the iterative approach is, in general, superior in speed and economy of storage.

At each step Eq. (28) requires a single-variable optimization to find ξ_n^*. Fortunately, the value of ξ_n^* can be determined analytically. It is shown in Appendix C that this value is given by

$$\xi_n^* = -\frac{\nabla h(\mathbf{x}_n) \cdot \nabla h(\mathbf{x}_n)}{|\mathbf{A} \nabla h(\mathbf{x}_n)|^2} \tag{30}$$

3. Increment n and use fixed-point Eq. (23) to update the state probabilities and link flows according to to the schemata of Figs. 6 and 7:

$$(\boldsymbol{p}_n, \boldsymbol{\lambda}_n) = \boldsymbol{G}(\boldsymbol{p}_{n-1}, \boldsymbol{\lambda}_{n-1})$$

4. Compute the current estimate of the mean transport time using the schema of Fig. 5:

$$\mathrm{E}[T_n] = F(\boldsymbol{p}_n, \boldsymbol{\lambda}_n)$$

5. If $|\mathrm{E}[T_n] - \mathrm{E}[T_{n-1}]| < \epsilon$, then halt, else go to step 3.

In steps 1 and 2 of the algorithm, we pretend that the network has buffers of unlimited capacity. Eqs. (24) and (26) define the state probabilities $\tilde{p}_m^{(ik)}$ and link flows $\tilde{\lambda}_{lmn:t}$ when infinite buffers are provided. Step 1 computes the link flows, and step 2 uses these link flows to compute the probability that K (the actual buffer size of the network under analysis) or more buffers are occupied in the hypothetical infinite-buffer network. As we shall see, these infinite-buffer quantities are easy to compute but serve only as a crude approximation to the corresponding quantities in the K-buffer network. These approximations are, however, sufficient to initialize the algorithm. Using these initial estimates, we then set the buffer size to K and iterate steps 3 through 5. The first iteration uses $\tilde{\lambda}_{lmn:t}$ as the network's link flows and $\tilde{p}_{\geq K}^{(ik)}$ as the buffer-overflow probabilities in the K-buffer network. Of course, these are only guesses, because they are derived from the infinite-buffer network. When successive iterations result in delay values that are indistinguishable, the algorithm terminates itself.

A central difficulty in solving sets of nonlinear equations, such as those that characterize our problem, is finding an appropriate starting solution. Step 1 of the algorithm computes an initial solution $\boldsymbol{\lambda}_0$ by solving the first-order linear approximation in Eq. (26). This effectively avoids the problem of solving a set of nonlinear equations, as is done in [16] and [18]. The algorithm clearly spends much of its time in step 1, solving the linear equations of Eq. (26). As written above, Eq. (26) includes exactly N^2 equations in the $9N^2$ unknowns $\tilde{\lambda}_{ijk:t}$ ($1 \leq i \leq N$, $0 \leq j \leq 2$, $0 \leq k \leq 2$, and $1 \leq t \leq N$). [Each individual equation corresponds to a tuple (i,t), where $1 \leq i \leq N$ and $1 \leq t \leq N$.] However, it may be observed that

$$\tilde{\lambda}_{ijk:t} = \begin{cases} \gamma_{it} & \text{if } k = \pi_1(i,t) \text{ and } j = 2 \\ 0 & \text{if } k \neq \pi_1(i,t) \end{cases}$$

for $1 \leq i \leq N$, $0 \leq j \leq 2$, $0 \leq k \leq 2$, and $1 \leq t \leq N$. The first part of this last equation holds for $3N^2$ entries, while the second part holds for $4N^2$ entries. Therefore, at least $7N^2$ of the variables $\tilde{\lambda}_{ijk:t}$ are trivially known in advance. Furthermore, any variable $\tilde{\lambda}_{lmn:t}$ for which $(i,k) \rightarrow (l,m)$ and $k = \pi_2(i,t)$ must be equal to 0, since no t-packets can flow over internal link (l,m,n). This further reduces the number of unknowns by N^2. So we have in Eq. (26) a set of N^2 linear equations in N^2 unknowns.

The system in Eq. (26) is sparse, as only about N of the N^2 unknowns appear in each equation. We can rewrite Eq. (26) in matrix form, as follows:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{27}$$

$$= \tilde{\lambda}_{i0k:t} + \tilde{\lambda}_{i1k:t} + \gamma_{it} - \gamma_{it} \sum_{\pi_1(i,s)=k} \tilde{\lambda}_{i0k:s} - \gamma_{it} \sum_{\pi_1(i,r)=k} \tilde{\lambda}_{i1k:r} + \gamma_{it} \left(\sum_{\pi_1(i,s)=k} \tilde{\lambda}_{i0k:s} \right) \left(\sum_{\pi_1(i,r)=k} \tilde{\lambda}_{i1k:r} \right)$$

Dropping the last term of this equation, which contains second-degree products of $\tilde{\lambda}$, we obtain as a first-order approximation the following linear equations:

$$\tilde{\lambda}_{lmn:t} \approx \tilde{\lambda}_{i0k:t} + \tilde{\lambda}_{i1k:t} + \gamma_{it} - \gamma_{it} \sum_{\pi_1(i,s)=k} \tilde{\lambda}_{i0k:s} - \gamma_{it} \sum_{\pi_1(i,r)=k} \tilde{\lambda}_{i1k:r} \tag{26}$$

for all i, k, l, m, n, t such that $1 \leq i \leq N$, $k = \pi_1(i,t)$, $(i,k) \rightarrow (l,m)$, $n = \pi_1(l,t)$, and $1 \leq t \leq N$.

Eq. (26) is the starting point for an algorithm to compute link flows in a deflection-routing network of K-buffer nodes. The idea is to use the $\tilde{\lambda}_{ijk:t}$ of Eq. (26) to initialize the algorithm. The infinite-buffer solution $\tilde{\lambda}_{ijk:t}$ is used as the initial approximation to the K-buffer solution $\lambda_{ijk:t}$. Given this initial solution, we can also compute initial values of $p_K^{(ik)}$ by appeal to Eq. (25), i.e., we approximate the probability of overflow in the K-buffer system by the probability that K or more of the infinite buffers are occupied. The algorithm then iteratively refines the solution vector $(\boldsymbol{\lambda}, \boldsymbol{p})$ by repeatedly invoking Eq. (23). The mean transport delay E$[T]$, which is a function of $(\boldsymbol{\lambda}, \boldsymbol{p})$, can then be computed by applying the schema of Fig. 5. The algorithm is specified below.

Input: Network topology $(i,k) \rightarrow (l,m)$ for all i,k; offered load (γ_{st}) for all s,t; and tolerance ϵ.

Output: Approximate mean delay E$[T]$ and internal-link flows $\boldsymbol{\lambda}$.

1. For all i, t assign

   $$\tilde{\lambda}_{i2k:t} = \gamma_{it}$$

   where $k = \pi_1(i,t)$, and solve the linear system in Eq. (26) for the initial link-flow vector:

   $$\boldsymbol{\lambda}_0 = (\tilde{\lambda}_{ijk:t})$$

   for all i, j, k, t. (Under very high traffic load the system might not have a solution; in this case we must use the initialization procedure described below.)

2. Compute the initial state probabilities according to Eq. (25) with $j = K$:

   $$\tilde{p}_{\geq K}^{(ik)} = \sum_{m=K}^{\infty} \tilde{p}_m^{(ik)} = \frac{(\tilde{\lambda}_{i0k}\tilde{\lambda}_{i1k})^{K-1}}{\Big[(1-\tilde{\lambda}_{i0k})(1-\tilde{\lambda}_{i1k})(1-\tilde{\lambda}_{i2k})\Big]^{K-2}}$$

   and

   $$\boldsymbol{p}_0 = (\tilde{p}_{\geq K}^{(ik)})$$

   for all i, k.

5 A procedure to calculate mean transport delay

In this section we develop a procedure that allows us to calculate the mean transport time, as given by Eq. (19). As a side effect the procedure also calculates link flows and state probabilities. By applying Eqs. (20) and (21) the network throughput can then be calculated also.

We first examine the extreme case of infinite-buffer stations, i.e., $K = \infty$. We denote by $\tilde{\lambda}_{ijk}$ and $\tilde{p}_m^{(ik)}$ the link flows and state-probability distributions when $K = \infty$. We also define $\tilde{\alpha}_j^{(ik)}$ analogously to Eq. (1), but using $\tilde{\lambda}$ instead of λ. Since there are unlimited packet buffers, there will never be packet deflections, i.e., $\delta_{ik} = 0$. To guarantee the stability of the Markov chain in Fig. 3 when $K = \infty$, we assume that $\tilde{\alpha}_0^{(ik)} > \tilde{\alpha}_6^{(ik)} + \tilde{\alpha}_7^{(ik)}$. Hence $\tilde{\alpha}_0^{(ik)} > \tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)}$, and we observe that

$$\lim_{K\to\infty} \frac{(\tilde{\alpha}_0^{(ik)})^K - (\tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)})^K}{(\tilde{\alpha}_0^{(ik)})^K} = 1 - \lim_{K\to\infty} \left(\frac{\tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)})}{\tilde{\alpha}_0^{(ik)}} \right)^K = 1$$

We can apply this observation to Eqs. (12) through (14), in which the reciprocal of the above term (with $\bar{\delta}_{ik} = 1$) appears as a factor. This yields the following formulas for the equilibrium-state probability distribution of L when $K = \infty$:

$$\begin{aligned} \tilde{p}_0^{(ik)} &= \tilde{\alpha}_0^{(ik)} - \tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)} \\ \tilde{p}_1^{(ik)} &= (1 - \tilde{\alpha}_0^{(ik)})(\tilde{\alpha}_0^{(ik)} - \tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)})/\tilde{\alpha}_0^{(ik)} \\ \tilde{p}_m^{(ik)} &= (\tilde{\alpha}_0^{(ik)} - \tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)})(\tilde{\alpha}_6^{(ik)} + \tilde{\alpha}_7^{(ik)})^{m-1}/(\tilde{\alpha}_0^{(ik)})^{m-1} \end{aligned} \tag{24}$$

for $m > 1$. This result may also be derived by solving the infinite-state Markov chain that corresponds to the finite chain of Fig. 3 as $K \to \infty$.

Since $\tilde{\alpha}_0^{(ik)} > \tilde{\alpha}_6^{(ik)} + \tilde{\alpha}_7^{(ik)}$,

$$p_\infty^{(ik)} \triangleq \lim_{m\to\infty} \tilde{p}_m^{(ik)} = \lim_{m\to\infty} (\tilde{\alpha}_0^{(ik)} - \tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)})(\tilde{\alpha}_6^{(ik)} + \tilde{\alpha}_7^{(ik)})^{m-1}/(\tilde{\alpha}_0^{(ik)})^{m-1} = 0$$

as one would expect. From Eqs. (1) and (24) we see that the probability of finding j ($j \geq 2$) or more packets at center $\mathcal{Q}_{ik}$ is given by

$$\begin{aligned} \tilde{p}_{\geq j}^{(ik)} \triangleq \sum_{m=j}^{\infty} \tilde{p}_m^{(ik)} &= \sum_{m=j}^{\infty} (\tilde{\alpha}_0^{(ik)} - \tilde{\alpha}_6^{(ik)} - \tilde{\alpha}_7^{(ik)}) \left(\frac{\tilde{\alpha}_6^{(ik)} + \tilde{\alpha}_7^{(ik)}}{\tilde{\alpha}_0^{(ik)}} \right)^{m-1} \\ &= \tilde{\alpha}_0^{(ik)} \left(\frac{\tilde{\alpha}_6^{(ik)} + \tilde{\alpha}_7^{(ik)}}{\tilde{\alpha}_0^{(ik)}} \right)^{j-1} \\ &= \frac{(\tilde{\lambda}_{i0k}\tilde{\lambda}_{i1k})^{j-1}}{\left[(1 - \tilde{\lambda}_{i0k})(1 - \tilde{\lambda}_{i1k})(1 - \tilde{\lambda}_{i2k})\right]^{j-2}} \end{aligned} \tag{25}$$

When $K = \infty$, $(i,k) \to (l,m)$, $n = \pi_1(l,t)$, and $k = \pi_1(i,t)$, Eq. (17) becomes

$$\tilde{\lambda}_{lmn:t} = \tilde{\lambda}_{i0k:t} + \tilde{\lambda}_{i1k:t} + \gamma_{it} \left(1 - \sum_{\pi_1(i,s)=k} \tilde{\lambda}_{i0k:s} \right) \left(1 - \sum_{\pi_1(i,r)=k} \tilde{\lambda}_{i1k:r} \right)$$

where an expression for $\mathrm{E}[L_{ik}]$ is found in Eq. (15), and

$$\begin{aligned}\lambda_{i2} &= \lambda_{i02:i} - \delta_{i02:i} + \lambda_{i12:i} - \delta_{i12:i} \\ &= \lambda_{i02} + \lambda_{i12} - p_K^{(i2)} \lambda_{i02} \lambda_{i12} \end{aligned} \tag{20}$$

is the rate at which packets exit station i via the user output port. The network throughput λ is then given by

$$\lambda = \sum_{i=1}^{N} \lambda_{i2} \tag{21}$$

We can also express in terms of the model's given parameters the probability that a packet from the user is discarded. Recall that the final term in the first part of Eq. (17) represents the rate at which t-packets from the user are admitted to $\mathcal{Q}_{ik}$. If $k = \pi_1(i,t)$, then the fraction of t-packets admitted to $\mathcal{Q}_{ik}$ is

$$\frac{\lambda_{i2k:t} \bar{\lambda}_{i0k} \bar{\lambda}_{i1k} \bar{\delta}_{ik}}{\lambda_{i2k:t}} = \bar{\lambda}_{i0k} \bar{\lambda}_{i1k} \bar{\delta}_{ik}$$

and, hence, the fraction of t-packets discarded at the user input port is

$$1 - \bar{\lambda}_{i0k} \bar{\lambda}_{i1k} \bar{\delta}_{ik} \tag{22}$$

We therefore see that the loss probabilities are the same for every t-packet offered to $\mathcal{Q}_{ik}$, if $k = \pi_1(i,t)$. The loss probability at each station should be kept below a fixed threshold, i.e., the value in Eq. (22) must never exceed the threshold for any i and k.

Our goal is to determine the mean transport time from Eq. (19) [and similarly, the network throughput from Eq. (21)], but we cannot do so without first determining the values of the variables in terms of which Eq. (19) is expressed. Fig. 5 shows the relationship among the variables of the various equations. The tree of Fig. 5 represents variables as nodes and labels arcs with equation numbers that appear in the text of the paper. If a node with label x has a directed arc labeled (n) to another node with label y, this means that the values of x can be determined by substituting the value for y into Eq. (n). A variable in the tree can be expressed in terms of variables that appear at the leaves of its subtree. Thus, in Fig. 5 the value of $\mathrm{E}[T]$ can ultimately be expressed in terms of the variables $\lambda_{ijk:t}$ and $p_K^{(ik)}$. We can therefore write $\mathrm{E}[T]$ as a function of $\lambda_{ijk:t}$ and $p_K^{(ik)}$: $\mathrm{E}[T] = F(\boldsymbol{p}, \boldsymbol{\lambda})$, where $\boldsymbol{p}$ is defined as the $N \times 3$ matrix $(p_K^{(ik)})$ of buffer-overflow probabilities, and $\boldsymbol{\lambda}$ is defined as the $N \times 3 \times 3 \times N$ matrix $(\lambda_{ijk:t})$ of internal-link flows.

The substitution tree of Fig. 6 specifies $p_K^{(ik)}$ as a function of $\lambda_{ijk:t}$ and $p_K^{(ik)}$, i.e., $\boldsymbol{p} = \boldsymbol{f}(\boldsymbol{p}, \boldsymbol{\lambda})$. We can also see from the substitution tree of Fig. 7 that $\lambda_{ijk:t}$ can be expressed in terms of $p_K^{(ik)}$ and $\lambda_{ijk:t}$, i.e., $\boldsymbol{\lambda} = \boldsymbol{g}(\boldsymbol{p}, \boldsymbol{\lambda})$. Defining $\boldsymbol{G}(\cdot)$ in terms of $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$, we have the following fixed-point vector equation:

$$(\boldsymbol{p}, \boldsymbol{\lambda}) = [\boldsymbol{f}(\boldsymbol{p}, \boldsymbol{\lambda}),\, \boldsymbol{g}(\boldsymbol{p}, \boldsymbol{\lambda})] \stackrel{\Delta}{=} \boldsymbol{G}(\boldsymbol{p}, \boldsymbol{\lambda}) \tag{23}$$

We can use the principle of iteration to solve for the values of $\boldsymbol{p}$ and $\boldsymbol{\lambda}$ that satisfy Eq. (23). Choosing an initial value $(\boldsymbol{p}_0, \boldsymbol{\lambda}_0)$, we define $(\boldsymbol{p}_{n+1}, \boldsymbol{\lambda}_{n+1}) \stackrel{\Delta}{=} \boldsymbol{G}(\boldsymbol{p}_n, \boldsymbol{\lambda}_n)$ for $n = 0, 1, \ldots$. If $(\boldsymbol{p}_0, \boldsymbol{\lambda}_0)$ is chosen carefully, then the sequence $(\boldsymbol{p}_0, \boldsymbol{\lambda}_0), (\boldsymbol{p}_1, \boldsymbol{\lambda}_1), \ldots$ converges to a value $(\boldsymbol{p}, \boldsymbol{\lambda})$ that satisfies Eq. (23). Given the value of $\boldsymbol{\lambda}$ we may then solve for the network transport delay and throughput using Eq. (19) and (21).

of 2 in the final term accounts for the use of a coin toss to resolve port contention between equal-priority packets.) The probability of deflection can now be written as

$$\delta_{ijk:t} = p_K^{(ik)} \lambda_{ijk:t} \left(\sum_{\pi_1(i,s)=k}^{p_i(s)>p_i(t)} \lambda_{i\bar{j}k:s} + \sum_{\pi_1(i,r)=k}^{p_i(r)=p_i(t)} \lambda_{i\bar{j}k:r}/2 \right) \tag{16}$$

(If $K = 1$, then clearly $p_K^{(ik)} = 1$ so that Eq. (16) corresponds to the expression for $\delta_{ijk:t}$ derived in [18].)

In deriving the expression of Eq. (16), we assume that the arriving t-packet finds the system full in the previous time slot with probability $p_K^{(ik)}$. That the state-probability distribution at arrival epochs is the same as the equilibrium-state probability distribution is not established and is, in all likelihood, false; this is because the interarrival process at queueing center $\mathcal{Q}_{ik}$ is not in general characterized by a geometric random variable. Thus, we cannot appeal to the so-called GASTA (geometric arrivals see time averages) property [23] to justify our use of the equilibrium-state probability $p_K^{(ik)}$ in Eq. (16). If we postulate that the arrival of a t-packet at input port j of station i occurs during a given time slot with probability $\lambda_{ijk:t}$ independently of other arrivals, then the arriving t-packet does indeed take a "random look" at queueing center $\mathcal{Q}_{ik}$. The degree to which this independence assumption holds affects the accuracy of our model. As long as there is adequate mixing of traffic streams, the arrival epochs of the t-packets are sufficiently randomized to justify the use of this independence assumption.

Referring to Fig. 4, we see that the principle of conservation of flow applies at every queueing center. If $(i,k) \rightarrow (l,m)$, then the flow of t-packets out of $\mathcal{Q}_{ik}$ must balance with the flow of t-packets into $\mathcal{Q}_{ik}$'s neighbor's internal link (l,m,n), where $0 \leq m \leq 1$ and $n = \pi_1(l,t)$:

$$\lambda_{lmn:t} = \begin{cases} \lambda_{i0k:t} - \delta_{i0k:t} + \lambda_{i1k:t} - \delta_{i1k:t} + \gamma_{it}\bar{\lambda}_{i0k}\bar{\lambda}_{i1k}\bar{\delta}_{ik} & \text{if } k = \pi_1(i,t) \\ \delta_{ik:t} & \text{if } k = \pi_2(i,t) \end{cases} \tag{17}$$

The term $\lambda_{ijk:t} - \delta_{ijk:t}$ is the effective rate of (undeflected) t-flow into $\mathcal{Q}_{ik}$ from input port j. Similarly, a packet from input port 2 is accepted by $\mathcal{Q}_{ik}$ only if there are no simultaneous arrivals or deflections from input ports 0 and 1; the probability of this is given by the final term in the first part of Eq. (17). The second part of Eq. (17) is the rate at which deflected t-packets enter center $\mathcal{Q}_{ik}$. Furthermore, the rate of flow on the internal links depends on the rate at which the user generates packets:

$$\lambda_{i2k:t} = \gamma_{it}$$

if $0 \leq k \leq 1$ and $\pi_1(i,t) = k$, and hence

$$\lambda_{i2k} = \sum_{\pi_1(i,t)=k} \gamma_{it} \tag{18}$$

for $0 \leq k \leq 1$. It is also clear that $\lambda_{ijk} \leq 1$ for $1 \leq i \leq N$, $0 \leq j \leq 2$, and $0 \leq k \leq 2$.

Let λ_{ik} be the rate at which packets pass through center $\mathcal{Q}_{ik}$. Since $\sum_i \sum_k L_{ik}$ is the expected number of packets in the entire network and $\sum_i \lambda_{i2}$ is the total throughput of the network, we can apply Little's result [24] to derive the following expression for the mean transport time:

$$\mathrm{E}[T] = \frac{\sum_{i=1}^{N} \sum_{k=0}^{2} \mathrm{E}[L_{ik}]}{\sum_{i=1}^{N} \lambda_{i2}} \tag{19}$$

general false that the four traffic streams behave in the network as in the isolated queueing center, e.g. the first three streams are not necessarily independent, as we have assumed in the isolated queueing model. This independence assumption obviously affects the accuracy of our model, but if each arrival class is comprised of a mixture of packets with different destinations, then the assumption will not have such a serious impact.

The rates of the four traffic streams entering queueing center $\mathcal{Q}_{ik}$ are λ_{i0k}, λ_{i1k}, λ_{i2k}, and δ_{ik}. Equating λ_{ijk} with λ_j and δ_{ik} with δ, we can define $\alpha_j^{(ik)}$ in analogy with the unsuperscripted definitions of Eq. (1). Now we can express the probability distribution of L_{ik}, the number of packets in $\mathcal{Q}_{ik}$, by assuming that the state of $\mathcal{Q}_{ik}$ is independent of the state of any other queueing center. Eqs. (4) through (6) become

$$p_0^{(ik)} = \frac{(\bar{\delta}_{ik}\alpha_0^{(ik)} - \alpha_6^{(ik)} - \alpha_7^{(ik)})(\bar{\delta}_{ik}\alpha_0^{(ik)})^K}{(\bar{\delta}_{ik}\alpha_0^{(ik)})^K - (\alpha_6^{(ik)} + \alpha_7^{(ik)})^K} \tag{12}$$

$$p_1^{(ik)} = \frac{(1-\bar{\delta}_{ik}\alpha_0^{(ik)})(\bar{\delta}_{ik}\alpha_0^{(ik)} - \alpha_6^{(ik)} - \alpha_7^{(ik)})(\bar{\delta}_{ik}\alpha_0^{(ik)})^{K-1}}{(\bar{\delta}_{ik}\alpha_0^{(ik)})^K - (\alpha_6^{(ik)} + \alpha_7^{(ik)})^K} \tag{13}$$

$$p_m^{(ik)} = \frac{(\bar{\delta}_{ik}\alpha_0^{(ik)} - \alpha_6^{(ik)} - \alpha_7^{(ik)})(\bar{\delta}_{ik}\alpha_0^{(ik)})^{K-m}(\alpha_6^{(ik)} + \alpha_7^{(ik)})^{m-1}}{(\bar{\delta}_{ik}\alpha_0^{(ik)})^K - (\alpha_6^{(ik)} + \alpha_7^{(ik)})^K} \tag{14}$$

for $2 \le m \le K$, and Eq. (8) becomes

$$\begin{aligned} \mathrm{E}[L_{ik}] = {} & \Big[(1-\bar{\delta}_{ik}\alpha_0^{(ik)})(\bar{\delta}_{ik}\alpha_0^{(ik)})^{K+1} + 2(\bar{\delta}_{ik}\alpha_0^{(ik)})^{K+1}(\alpha_6^{(ik)} + \alpha_7^{(ik)}) - \\ & (\bar{\delta}_{ik}\alpha_0^{(ik)})^K(\alpha_6^{(ik)} + \alpha_7^{(ik)})^2 - (K+1)\bar{\delta}_{ik}\alpha^{(ik)}(\alpha_6^{(ik)} + \alpha_7^{(ik)})^K + \\ & K(\alpha_6^{(ik)} + \alpha_7^{(ik)})^{K+1}\Big] / \Big[(\bar{\delta}_{ik}\alpha_0^{(ik)})^K - (\alpha_6^{(ik)} + \alpha_7^{(ik)})^K\Big](\bar{\delta}_{ik}\alpha_0^{(ik)} - \alpha_6^{(ik)} - \alpha_7^{(ik)}) \end{aligned} \tag{15}$$

To derive an expression for $\delta_{ijk:t}$ in terms of t-flows, we observe that the deflection of a t-packet away from internal link (i,j,k) toward internal link (i,j,k') during time slot n, where $k' = \pi_2(i,t)$, depends upon the joint occurrence of the following events:

- $\mathcal{A} \triangleq$ one buffer of $\mathcal{Q}_{ik}$ is free in time slot n
- $\mathcal{B} \triangleq$ a t-packet arrives at internal link (i,j,k) in time slot n
- $\mathcal{C}_s \triangleq$ an s-packet with priority greater than $p_i(t)$ arrives at internal link $(i,\bar{j},k)$ in time slot n
- $\mathcal{D}_r \triangleq$ an r-packet with priority $p_i(t)$ arrives at internal link $(i,\bar{j},k)$ in time slot n

The event $\mathcal{A}$ is equivalent to the event that all K buffers of $\mathcal{Q}_{ik}$ are occupied in time slot $n-1$. When n is large, $\mathcal{Q}_{ik}$ is in equilibrium, and hence $\mathrm{P}[\mathcal{A}] = p_K^{(ik)}$. We can thus express the probabilities of events $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}_s$, and $\mathcal{D}_r$ as

$$\begin{aligned} \mathrm{P}[\mathcal{A}] &= p_K^{(ik)} \\ \mathrm{P}[\mathcal{B}] &= \lambda_{ijk:t} \\ \mathrm{P}[\mathcal{C}_s] &= \lambda_{i\bar{j}k:s} \qquad \text{if } p_i(s) > p_i(t) \\ \mathrm{P}[\mathcal{D}_r] &= \lambda_{i\bar{j}k:r} \qquad \text{if } p_i(r) = p_i(t) \end{aligned}$$

Assuming the independence of these events, we can write the probability that a t-packet is deflected from internal link (i,j,k) at the beginning of time slot n as $\mathrm{P}[\mathcal{A}]\mathrm{P}[\mathcal{B}](\mathrm{P}[\cup_s \mathcal{C}_s] + \mathrm{P}[\cup_r \mathcal{D}_r]/2)$. (The denominator

Since the αs are defined entirely in terms of λ_0, λ_1, and λ_2, we have derived an expression for the mean queue length that depends only on the arrival rates to the queueing center. In the next section we address the problem of determining the values of these arrival rates when the queueing centers are connected together and then analyze the performance of the interconnected centers.

4 Network analysis

In this section we derive an expression for the mean transport time E[T] and the network throughput λ. Whereas in the previous section we analyzed the performance of an isolated queueing center, in this section we study the performance of a network of queueing centers. Recall from the previous section that the state-probability distribution can be computed from Eqs. (4) through (6), given the parameters λ_0, λ_1, λ_2, and δ. To analyze the network we treat it as a collection of independent queueing centers and derive values for the parameters that correspond to λ_0, λ_1, λ_2, and δ at each center.

We define a *t-packet* to be a packet with destination t. We also call that component of an internal link's flow destined for station t the *t-flow* on the link. Define $\lambda_{ijk:t}$ to be the probability that a t-packet arrives at input port j of station i, where $k = \pi_1(i,t)$. Recall that λ_{ijk} is the rate of flow on internal link (i,j,k), and so clearly

$$\lambda_{ijk} = \sum_{t=1}^{N} \lambda_{ijk:t} \tag{9}$$

We define $\delta_{ijk:t}$ to be the probability that a t-packet is deflected away from internal link (i,j,k), and $\delta_{ik:t}$ and δ_{ik} to be the rate at which deflected t-packets and packets, respectively, flow into center $\mathcal{Q}_{ik}$. A t-packet deflected into center $\mathcal{Q}_{ik}$ can come from either center $\mathcal{Q}_{i\bar{k}}$ or $\mathcal{Q}_{i2}$, and output port k must be the secondary preference for the packet. Therefore,

$$\delta_{ik:t} = \begin{cases} \delta_{i0\bar{k}:t} + \delta_{i1\bar{k}:t} + \delta_{i02:t} + \delta_{i12:t} & \text{if } k = \pi_2(i,t) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

and

$$\delta_{ik} = \sum_{\pi_2(i,t)=k} \delta_{ik:t} \tag{11}$$

for $0 \le k \le 2$.[‡] We note that the use of the sum in Eq. (11) is valid, since there never are more than one packet deflection per time slot in a station.

We now apply the results of the previous section to analyze the network of queueing centers $\mathcal{Q}_{ik}$. The queuing model of the previous section assumes that all four classes of traffic entering the queueing center behave as Bernoulli processes of intensities λ_0, λ_1, λ_2, and δ. Similarly, each queueing center $\mathcal{Q}_{ik}$ accepts traffic from four points within the station, i.e., from input ports 0, 1, and 2, and from any point that deflects packets to it. We therefore assume that these streams also behave as Bernoulli processes. However, it is in

[‡]The index of summation in Eq. (11) is t, and the summation is over all values of t that satisfy the condition $\pi_2(i,t) = k$; similar notation will also be used throughout the remainder of the paper.

are given by the following equations:

$$\begin{aligned}
\mathbf{P}_{0,0} &= \bar{\delta}\alpha_0 \\
\mathbf{P}_{0,1} &= 1-\bar{\delta}\alpha_0-\alpha_6-\alpha_7 \\
\mathbf{P}_{0,2} &= \alpha_6+\alpha_7 \\
\mathbf{P}_{m,m-1} &= \bar{\delta}\alpha_0 && \text{if } 1 \le m \le K \\
\mathbf{P}_{m,m} &= 1-\bar{\delta}\alpha_0-\alpha_6-\alpha_7 && \text{if } 1 \le m < K \\
\mathbf{P}_{m,m+1} &= \alpha_6+\alpha_7 && \text{if } 1 \le m < K \\
\mathbf{P}_{K,K} &= 1-\bar{\delta}\alpha_0
\end{aligned}$$

We let $p_m^{(n)}$ be the probability that the queueing center contains m packets in its buffers during time slot n. Define $p_m \triangleq \lim_{n\to\infty} p_m^{(n)}$ to be the probability that the center contains m packets in its buffers when the center is in equilibrium, and let $\mathbf{p} \triangleq (p_0, p_1, \ldots, p_K)$ be the equilibrium-state probability vector, which is the solution to the following equations:

$$\mathbf{p} = \mathbf{p}\mathbf{P} \tag{2}$$

and

$$p_0 + p_1 + \cdots + p_K = 1 \tag{3}$$

The solution to Eqs. (2) and (3), which describe a Markov chain that is nearly a birth–death process (see Fig. 3), is easily obtained. In Appendix A we show that the vector $\mathbf{p}$ that satisfies Eqs. (2) and (3) is given by the following equations:

$$p_0 = \frac{(\bar{\delta}\alpha_0-\alpha_6-\alpha_7)(\bar{\delta}\alpha_0)^K}{(\bar{\delta}\alpha_0)^K-(\alpha_6+\alpha_7)^K} \tag{4}$$

$$p_1 = \frac{(1-\bar{\delta}\alpha_0)(\bar{\delta}\alpha_0-\alpha_6-\alpha_7)(\bar{\delta}\alpha_0)^{K-1}}{(\bar{\delta}\alpha_0)^K-(\alpha_6+\alpha_7)^K} \tag{5}$$

$$p_m = \frac{(\bar{\delta}\alpha_0-\alpha_6-\alpha_7)(\bar{\delta}\alpha_0)^{K-m}(\alpha_6+\alpha_7)^{m-1}}{(\bar{\delta}\alpha_0)^K-(\alpha_6+\alpha_7)^K} \tag{6}$$

for $2 \le m \le K$.

Define the random variable L to be the number of packets in the isolated queueing center. The mean of L is given by

$$\mathrm{E}[L] = \sum_{m=1}^{K} m p_m \tag{7}$$

Substituting Eqs. (5) and (6) into Eq. (7) and simplifying, we obtain the following formula for the expected number of packets in the isolated queueing center:

$$\begin{aligned}
\mathrm{E}[L] = {} & \Big[(1-\bar{\delta}\alpha_0)(\bar{\delta}\alpha_0)^{K+1} + 2(\bar{\delta}\alpha_0)^{K+1}(\alpha_6+\alpha_7) - \\
& (\bar{\delta}\alpha_0)^K(\alpha_6+\alpha_7)^2 - (K+1)\bar{\delta}\alpha_0(\alpha_6+\alpha_7)^K + \\
& K(\alpha_6+\alpha_7)^{K+1}\Big] \Big/ \big\{[(\bar{\delta}\alpha_0)^K-(\alpha_6+\alpha_7)^K](\bar{\delta}\alpha_0-\alpha_6-\alpha_7)\big\}
\end{aligned} \tag{8}$$

It will be convenient to define the following symbols:

$$\left.\begin{array}{rcl} \alpha_0 & \triangleq & \bar{\lambda}_0\bar{\lambda}_1\bar{\lambda}_2 \\ \alpha_1 & \triangleq & \lambda_0\bar{\lambda}_1\bar{\lambda}_2 \\ \alpha_2 & \triangleq & \bar{\lambda}_0\lambda_1\bar{\lambda}_2 \\ \alpha_3 & \triangleq & \bar{\lambda}_0\bar{\lambda}_1\lambda_2 \\ \alpha_4 & \triangleq & \bar{\lambda}_0\lambda_1\lambda_2 \\ \alpha_5 & \triangleq & \lambda_0\bar{\lambda}_1\lambda_2 \\ \alpha_6 & \triangleq & \lambda_0\lambda_1\bar{\lambda}_2 \\ \alpha_7 & \triangleq & \lambda_0\lambda_1\lambda_2 \end{array}\right\} \qquad (1)$$

Each value of α_j is the joint probability of a particular pattern of class-0, -1, and -2 arrivals. In particular, the joint probabilities α_0 and α_3 correspond to events in which there are no class-0 and -1 arrivals in the given time slot, and these are precisely the conditions under which a class-3 packet may arrive. Hence, the joint probabilities $\delta\alpha_0$ and $\delta\alpha_3$ correspond to the event that a class-3 packet arrives during a given time slot. Ultimately, then, there are ten distinct arrival patterns of the four packet classes (the six forbidden patterns include events corresponding to such joint probabilities as $\lambda_0\lambda_1\lambda_2\delta$, etc.).

It is clear that the number of packets buffered at the isolated queueing center during the next time slot $n+1$ depends only on the number of packets buffered during the current time slot n and, given this current state, is statistically independent of the numbers buffered at earlier time slots $n-1, n-2, \ldots, 0$. So we model the queueing center as a discrete-time Markov chain, the time-dependent state of which is defined to be the number of packets buffered during a particular time slot.

Because there are so many arrival classes, the Markov chain is not easily specified. Rather than jumping immediately to a description of the one-step–transition probability matrix of the chain, we first examine the behavior of the chain on a state-by-state basis.

Fig. 2 illustrates the one-step transitions that can occur in the chain. For each state of the chain (viz. $0, 1, \ldots, K$), there are ten possible transitions out of the state, and these transitions correspond to the ten permitted arrival patterns of the four packet classes, i.e., α_1, α_2, α_4 through α_7, $\delta\alpha_0$, $\bar{\delta}\alpha_0$, $\delta\alpha_3$, and $\bar{\delta}\alpha_3$. For example, leaving state 1, the chain can enter state 2 via two mutually exclusive arrival patterns, i.e., those occurring with joint probabilities α_6 and α_7. Note also that even though α_7 corresponds to the simultaneous arrival of class-0, -1, and -2 packets, the class-2 packet is discarded, according to rule 3, above.

Using Fig. 2, we can easily construct the full state-transition graph for the Markov chain. The state-transition graph, shown in Fig. 3, is nearly a birth–death process, and thus the equilibrium-state probabilities are easy to solve. From Fig. 3 we see that the nonzero entries of $\mathbf{P}$, the one-step–transition probability matrix,

We view the isolated queueing center as a discrete-time, single-server queue with deterministic service in which there are four classes of packets. The queue contains buffers for at most K packets. During any given time slot class-0, -1, and -2 packets, may arrive at the queueing center with probabilities λ_0, λ_1, and λ_2, respectively. Class-3 packets, however, arrive only in the absence of class-0 and -1 arrivals; given the absence of class-0 and -1 arrivals, the probability of a class-3 arrival is δ. Hence, the first three arrival streams behave as independent Bernoulli processes, while the fourth arrival stream is clearly dependent on the first two streams but independent of the third stream.

The queueing center services arrivals from four sources, each of which is associated with a traffic class. The four traffic classes of the isolated queueing center $\mathcal{Q}_{ik}$ correspond to quasi-independent streams of traffic that feed into an output port of the station, i.e., those traffic streams coming into $\mathcal{Q}_{ik}$ from the two network input ports (classes 0 and 1), the user input port (class 2), and the other output ports as a result of deflection (class 3). Traffic classes 0 through 2 are statistically independent of each other. However, arrivals from traffic class 3 depend on classes 0 and 1. In Fig. 1, for example, center $\mathcal{Q}_{i0}$ is fed by class-0 traffic via the arc labeled λ_{i00}, by class-1 traffic via the arc labeled λ_{i10}, by class-2 traffic via the arc labeled λ_{i20}, and by class-3 traffic via the (dotted) arc from center $\mathcal{Q}_{i1}$.

Our hypothetical queueing center operates according to five rules:

1. During a given time slot class-0, -1, and -2 packets, respectively, are generated independently with probabilities λ_0, λ_1, and λ_2, respectively.

2. During a given time slot a class-3 packet is generated with probability δ, but only when no class-0 or -1 packet is generated, e.g. a class-2 packet and a class-3 packet arrive simultaneously with probability $\bar{\lambda}_0\bar{\lambda}_1\lambda_2\delta$.†

3. If a class-2 packet arrives simultaneously with another packet, then it is discarded.

4. When class-0 and -1 packets arrive simultaneously, and only a single buffer is available, then one is discarded and one is queued.

5. All service times are one time slot in duration.

†We define $\bar{x} \triangleq 1 - x$ to be the complement of x.

The order in which the two packets are queued is determined by the packets' priorities: if the packets have respective priorities p_1 and p_2, with $p_1 > p_2$, then the first packet is enqueued in front of the second.

Packets in transit are always served before packets from the user. If station i generates a packet with destination t during a time slot, then this packet will be placed on output port k, where $k = \pi_1(i,t)$, only if no other in-transit packet that arrives in the same time slot is routed through output port k. If another in-transit packet contends with the user packet—either because the in-transit packet is routed or deflected through output port k—then the user packet is discarded before it enters the network. If the user packet is accepted, it is placed on the end of the packet queue for output port k. This restrictive policy is rooted in the thinking that it is better to block packets from entering the network before they have the chance to cause congestion. Other access policies could be modeled by the techniques to be presented.

We assume that the time to assign priorities and route packets from an input to an output port is a negligible fraction of the time slot.

To simplify the analysis we ignore the effect of signal-propagation time, but this could be handled easily, as in [22] and [18].

A queueing model of the station is illustrated in Fig. 1. We refer to the three queueing centers in station i as $\mathcal{Q}_{i0}$, $\mathcal{Q}_{i1}$, and $\mathcal{Q}_{i2}$. Each queueing center $\mathcal{Q}_{ik}$ has K packet buffers uses the first-come–first-served discipline, and requires one time slot to serve a packet. We assume throughout the paper that at least two buffers are provided ($K \geq 2$), since the complete solution for single-buffer stations ($K = 1$) has already been obtained. The results in this paper may be extended easily to the case of $K = 1$, but the analysis and formulas will be slightly different from what appears in the sequel. Deflections and discards are handled as previously described, and the decision to deflect or discard a packet is made just before the packet enters the queueing center.

The arc from input port j to output port k of the station shown in Fig. 1 is called *internal link* (i,j,k), and λ_{ijk} represents the (predeflection) probability that a packet is found on internal link (i,j,k) at the start of a given time slot. The dotted lines depict the paths taken by packets that are deflected from their primary output ports to their secondary output ports.

In the next sections we first analyze the performance of an isolated queueing center that is structurally equivalent to $\mathcal{Q}_{ij}$ and then explore how these queueing centers behave when they are connected together. Ultimately, we determine the mean of T, the interval between an (undiscarded) packet's time of entry into its source station and time of exit from its destination station (i.e., the transport time).

3 Queueing-center analysis

In this section we analyze the queueing center $\mathcal{Q}_{ik}$ in isolation. We then derive formulas for the probability distribution of the number of packets in the isolated center, given a specific arrival process, as described below.

deflected to user output port 2, since this would result in its misdelivery). A packet destined for station t would normally be routed via output port $\pi_1(i,t)$ (its first preference). In case the packet were deflected, however, it would be routed via output port $\pi_2(i,t)$ (its second preference). Thus, the set of N vectors $\{[\pi_1(i,t), \pi_2(i,t)] : 1 \leq t \leq N\}$ can be viewed as the routing table for station i. We assume that the first-preference output port $\pi_1(i,t)$ indicates the start of the shortest path (in hops) to station t from station i. Some topologies have stations from which a shortest path to a given destination is via either output port, viz. "don't care" stations; in these cases we assume that each packet has a unique first-preference port.

Each packet that arrives at a station from the network (i.e., not from the user input port) is assigned a priority that can be used to resolve the contention that arises when two packets vie simultaneously for the last unoccupied buffer of the same output port. When two packets contend in this way, the packet with the highest priority is granted access to the port, while the other must be deflected to another output port (which is guaranteed to have an unoccupied buffer). Thus, if the contending packets have respective priorities p_1 and p_2, with $p_1 > p_2$, then the first packet is given access to the output port while the second is deflected. If both contending packets possess identical priority, then access is granted randomly by a coin toss. Although one can imagine several different priority schemes, we restrict ourselves to schemes in which priorities are assigned to a packet only on the basis of the assigning station's identity and the packet's destination; this motivates us to define an integer-valued priority function $p_i(t)$, where i is the assigning station and t is the destination of the packet.

We principally consider a priority scheme that gives a high priority to the packet that would be most penalized by a deflection. Such a strategy guarantees that whenever two packets contend for a common output port, the ensuing deflection produces the smallest number of additional hops to deliver both packets, if we assume that no other deflections will occur. Formally, if we define $d_1(i,t)$ and $d_2(i,t)$, respectively, as the shortest distance (in hops) from station i to station t by way of output port $\pi_1(i,t)$ and $\pi_2(i,t)$, respectively, then the scheme assigns the priority of $d_2(i,t) - d_1(i,t)$ to a packet that arrives at station i and is destined for station t, i.e., $p_i(t) = d_2(i,t) - d_1(i,t)$ for $1 \leq i \leq N$ and $1 \leq t \leq N$. By equating each packet's priority to the difference between their primary and alternate path lengths, we ensure that the highest priority is given to that packet that would suffer most from being deflected. In words, given the simultaneous arrival of two packets, this scheme chooses the optimal switching decision, in the sense that the total number of hops to all destinations is minimized, as in the so-called minimum-distance algorithm of [21]. It will be noted that these policies do not provide globally optimal solutions, since decisions are based only on information obtained locally at the switch. The analysis to be presented, however, allows for the function $p_i(t)$ to take any form. For example, defining $p_i(t) = -d_1(i,t)$ always gives precedence to the packet whose destination is nearest, while defining $p_i(t) = 0$ requires that contention be resolved by the toss of an unbiased coin.

If two in-transit packets arrive simultaneously and are to be routed through the same output port, then they are both placed on the end of the packet queue for the output port, if at least two buffers are unoccupied.

of linear equations—is substantial. Thus, the best that we currently hope for when evaluating deflection-routing networks under nonuniform traffic is to obtain an efficient algorithm to determine the value of the performance metric of interest.

This paper is organized as follows. The modeling assumptions are stated in Section 2. Section 3 analyzes the performance of an individual station's queueing centers in isolation. Section 4 uses the results of Section 3 to derive an expression for the network-wide mean transport time and throughput, given the link flows into the station. Section 5 presents an algorithm for calculating the rates of flow on the network's links and applying the formulas of Section 4 to obtain delay and throughput estimates for the network. In Section 6 our model is validated against simulations of the network, and results are discussed. We conclude with Section 7 by reviewing our results and proposing new areas for research in the analysis of deflection routing. Appendices provide proofs of key mathematical results and a list of the mathematical symbols used.

2 Assumptions and model

Our mathematical model describes an N-station network in which every station has two input ports from the network, a user input port, two output ports to the network, and a user output port. All $2N$ links of the network transmit at the same speed. Each of the three output ports has K packet buffers; thus, deflection routing is used when buffer overflow would occur. Each output port to the network is connected to exactly one input port, and each input port is connected to exactly one output port. The topology of such a network can be represented as a directed graph in which each node has two incoming and two outgoing arcs. We represent the topology of the network in terms of the interconnection between each station's ports, and we use the notation $(i, j) \rightarrow (l, m)$ to mean that output port j of station i is connected to input port m of station l.

Time is assumed to be divided into slots of a fixed duration, and all arrivals to and departures from the station are synchronized to occur on time-slot boundaries. Packets have a fixed length and require one time slot for transmission. During any time slot the user at station i may submit no more than one packet to the network, and the probability that this user submits a packet destined for station j is γ_{ij}. We assume that streams of packets between source–destination pairs are generated by users independently of each other, and that the intervals between successively generated packets are independent of each other. Our analysis will rely on the assumptions that each node of the network receives streams of packets that are independent of each other and that each stream consists of packets that have independent interarrival times. As shown in [16], these independence assumptions can be violated in actual networks. In practice, however, the independence assumptions have yielded results that conform well to actual network behavior.

Numbering the output ports from 0 to 2, as in Fig. 1, we define the *preference vector* $[\pi_1(i,t), \pi_2(i,t)]$ to specify the first and second preferences of a packet arriving at station i with destination station t, where the first component of the vector can range from 0 to 2 and the second from 0 to 1 (a packet is never

realizes significant cost savings over shortest-path routing used with conventional store-and-forward networks, which must provide ample packet buffers in each packet switch. This fact is important in the design and implementation of all-optical networks—which commonly rely on deflection routing—as delay-line–based optical buffers are costly to incorporate into a switch [1, 2]. Intuitively, the analysis of deflection-routing networks is difficult, because a packet can have an unbounded number of distinct routes from its source to its destination; moreover, the probability that a packet is deflected depends on the probability that there are output-link–access conflicts with other packets (which themselves might have been deflected). The goal of this paper is to present a mathematical model of multibuffer deflection-routing networks that can be used to provide good estimates of mean packet delay under arbitrary traffic demands.

The most-tractable performance models of deflection routing deal with networks operating under uniform traffic load. An early analysis of deflection routing appears in [3], which studies the Manhattan street network with single-buffer stations under uniform traffic conditions. An extension of [3] to the bidirectional Manhattan street network is presented in [4] and [5]. Both [6] and [7] present analyses of the performance of ShuffleNet with single-buffer stations under uniform traffic load. Performance evaluations of the Manhattan street network with multibuffer stations and uniform traffic are found in [8], [9], and [10], with the first reference obtaining the probability distribution of the transport time. References [11] and [12] emphasize the evaluation of evacuation time (the time to clear all packets from the network once packet input has ceased) in several classes of single-buffer deflection-routing networks under uniform traffic. In [13] both the transient and equilibrium delay is obtained for single-buffer deflection-routing hypercube networks under uniform traffic.

Progress has also been made in analyzing deflection-routing networks under nonuniform (arbitrary) traffic load. The work of [14] and [15] concentrates on analyzing the performance of deflection-routing networks in which stations have many input and output ports (e.g. multidimensional toroidal meshes and hypercubes). That work extends the analysis of these networks to specific classes of nonuniform (though symmetric) traffic but does not consider the case of arbitrarily nonuniform traffic. The probability distribution of the transport time in the Manhattan street network with single-buffer stations under nonuniform traffic is derived in [16] and [17]. In [18] the mean end-to-end packet time is derived for an arbitrary two-connected topology of single-buffer stations under nonuniform traffic conditions. Using mean-value analysis, [19] analyzes the mean end-to-end packet time in deflection-routing ShuffleNets with single-buffer stations and nonuniform traffic.

The first analysis of deflection routing under nonuniform traffic in networks with multibuffer stations appears in [20]. Building upon that work, in this paper we develop a model of multibuffer deflection-routing networks under nonuniform traffic and a procedure that allows us to solve the model for the mean transport time and network throughput.

Whereas analytical results for the case of uniform traffic are often closed-form expressions, all known analytical results for the case of nonuniform traffic require the solution of a set of equations. The computational effort to solve these equations—whether a modest set of nonlinear equations or an enormous set

Analyzing Deflection Routing under Nonuniform Traffic*

Joseph Bannister
The Aerospace Corporation
Los Angeles, California, USA

Flaminio Borgonovo
Luigi Fratta
Dipartimento di Elettronica
Politecnico di Milano, Italy

Mario Gerla
Computer Science Department
University of California at Los Angeles, USA

Abstract

Deflection routing can be used in networks whose stations have the same number of input and output links. Fixed-length packets arrive synchronously on the station's input links at the beginning of time slots, and each packet is routed via the output link that offers the shortest path to its destination. Since the number of packet buffers at each output link is finite, the simultaneous contention of two packets for the last buffer of a common output link must be resolved by "deflecting" one of the packets to another output link. Thus, the deflection of a packet could result in the packet following a route that is not a shortest path.

The potentially unbounded number of routes that a given packet can take makes analyzing the performance of such networks difficult. In particular, there currently are no analytical models that can analyze multibuffer deflection-routing networks with nonuniform traffic. Using independence assumptions, we develop a performance model of deflection routing that allows us to estimate accurately and efficiently the mean transport time and throughput in a network that has any given two-connected topology, multiple buffers at each output port, and an arbitrary traffic matrix.

1 Introduction

Networks in which all packet switches have the same number of input and output links can use the technique of deflection routing. With this technique, when two packets simultaneously require the same output link, the packet switch "deflects" one of the packets to an alternate output link. If packet arrivals to the packet switch are synchronized and all packets have a fixed length, then deflection routing permits the use of as few as one packet buffer per output link. Deflection routing's requirement for only a few packet buffers

*This work was partially supported by the Telecommunication Project of the CNR of Italy, US DARPA contract MDA903-87-C-0663, State of California–Pacific Bell MICRO grant 4-541121-19907 KC7F, and the Aerospace Sponsored Research program. An earlier version of this paper was presented at IEEE INFOCOM '92.