The Design and Analysis of a Semidynamic Deterministic Routing Rule

TAK-SHING P. YUM, MEMBER, IEEE

(Invited Paper)

***Abstract*—The best deterministic rule, newly proposed in this paper is similar in nature to the best stochastic rule [3], except that 1) a maximum traffic bifurcation flow distribution is chosen and 2) deterministic routing sequences are used. Analysis shows that the best deterministic rule always gives better delay performance than the best stochastic rule. A semidynamic version of this rule is introduced for use in a varying traffic rate environment.**

I. INTRODUCTION

THERE are four essential components of information that can be used by a routing rule in a computer-communication network. 1) The *topological information* concerns the entire network. It includes, for example, whether a given outgoing link can lead to a certain destination, number of hops from the originating node to each destination node, etc. It is changed whenever the network is expanded, some parts of it are removed, or when nodes and links fail. 2) The *traffic rate information* accounts for the external traffic intensity of each source and destination pair. It may change over a period of perhaps hours, in contrast to the topological information which changes perhaps over days or even weeks. As an example in one region, the traffic may peak at certain hours during the morning and afternoon, and may decline considerably at night. 3) The *local queue length information*, includes lengths of output queues, the types of messages in each queue, their priorities, etc., at each local node. 4) The *feedback information* includes the state of the queues and other local information at *neighboring* nodes. Usually, only portions of this are fed back and used.

Rules which incorporate feedback information are called *feedback rules*. Otherwise they are called *local rules*. Routing rules that use the instantaneous queue length information for their routing decisions are called *adaptive rules*. Otherwise they are called *fixed rules*. Adaptive rules have been shown to give better delay performance than the fixed rules, but their implementations are more complex. Fixed rules have been studied extensively in the literature [2]. Most of them are of the stochastic type, i.e., messages are distributed to the output buffers under a fixed probability assignment. One particular stochastic fixed rule that minimizes the average message delay is analyzed in [3]. We refer to it here as the BS (*best stochastic*) rule. We shall first summarize the BS rule in the next section because it is the substructure of the BD (*best deterministic*) rule, the rule to be studied here.

In Section III we introduce the flow distribution entropy function H. We shall see that finding the flow distribution of maximum H is essential for the efficient operation of the BD rule. We then introduce the BD rule and discuss how it operates in a network under a varying traffic rate environment. In Section IV, we study the properties of the BD rule and compare its delay performance to the BS rule.

The primary purpose of this paper is to single out the routing aspect of the network operation and demonstrate, through analysis, the improvement possible by using deterministic routing sequences to bifurcate traffic. We would also like to emphasize that this kind of theoretical study provides insight in "custom designing" routing rules for specific networks. It also points out the directions for improving network performance (more throughput, less delay) through the choice of routing rules.

Note that the BS rule is used primarily for network design purposes; there is no known implementation of it in any existing network. As for the BD rule, it is newly proposed. Deterministic routing sequences introduced here are being used in the Common Channel Interoffice Signaling (CCIS) network for telephone signaling. The use of the optimum flow distribution and the semidynamic version of the rule in the CCIS network is still under investigation.

II. THE BEST STOCHASTIC ROUTING RULE [3], [10]

The BS rule is globally optimum in the sense that it gives a minimum overall average time delay, averaged over all nodes of the network and averaged over statistically varying time delay due to traffic fluctuation, given that *traffic is bifurcated (or routed to different outgoing links) by fixed probability assignments at each node*. Here, we assume that the input traffic rates are fixed. In situation where traffic rates do change, this rule can be operated in a semidynamic mode with new routing decisions calculated periodically from new estimates of the traffic rates. Algorithms incorporating this feature can be found in [4], [5]. Let us consider a hypothetical computer communication network with N nodes and L links. Let C_i be the capacity of link i in bits/s; λ_i be the rate of average message flow over link i; $1/\mu$ the average length of exponentially distributed messages, assumed to be the same throughout the network; and γ_{ij} the external Poisson arrival rate of messages from node i to node j. After invoking the message length independence assumption [6],

Paper received April 20, 1979; revised November 11, 1980.
The author is with the Institute of Computer Engineering, National Chiao-Tung University, Hsinchu, Taiwan, Republic of China.

the complex queueing network is reduced to a network of $M/M/1$ queues. The average time delay is

$$\bar{T} = \frac{1}{\sum_{i,j} \gamma_{i,j}} \sum_{i=1}^{L} \left[\frac{f_i}{C_i - f_i} + f_i T_i \right] \tag{1}$$

where T_i is the nonqueueing delay in link i (e.g., processing delay, propagation delay, etc.), and $f_i \triangleq \lambda_i/\mu$ is the flow in link i. Our routing problem is similar to the multicommodity flow problem in network-flow theory. The objective is to assign paths for each commodity so as to minimize $\bar{T}$. Let there be M source-destination pairs (or commodities) in the network and let the average flow in link i due to commodity k be denoted as f_i^k. Then

$$f_i = \sum_{k=1}^{M} f_i^k \qquad i = 1, 2, \cdots, L \tag{2}$$

and the set $\{f_i^k\}$ completely specifies the routing strategy. There are two constraints on $\{f_i^k\}$. 1) The capacity constraints say that $f_i < C_i$ for all i. 2) the flow constraints say that message flows must be conserved, commodity by commodity. Thus, if we label commodities by the source-destination nodal pairs, and links by the two nodes to which they are connected, we have at node l, due to comodity (i, j) (from [10])

$$\sum_{k=1}^{N} f_{kl}^{ij} - \sum_{m=1}^{N} f_{lm}^{ij} = \begin{cases} -\gamma_{ij}/\mu & \text{if } l = i \\ \gamma_{ij}/\mu & \text{if } l = j \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

There are various algorithms for solving this constrained minimization problem. The flow deviation (FD) method [3] is the earliest one. The extremal flow (EF) method [7] and the gradient projection (GP) method [8] are more recent and execute in less time. They all use the iteration approach and rely on the fact that $\bar{T}$ is convex, as is the feasible set of multicommodity flows. Thus, a unique global minimum exists. The FD method provides the *total flow* $\{f_i\}$. The *individual commodity flows* $\{f_i^k\}$ required for routing assignment must be determined by additional "bookkeeping."

III. THE BEST DETERMINISTIC ROUTING RULE

The BD rule differs from the BS rule in two aspects: the choice of maximum traffic bifurcation and the use of deterministic routing sequences. In this section, we shall first elaborate on these two and then discuss the operation of the BD rule in a network.

A. Maximum Traffic Bifurcation

In the preceding section, we have indicated the use of the FD method, among others, to find the total flow $\{f_i\}$. Now for a specific set of f_i, say $\{f_i^*\}$, there corresponds many sets of $\{f_i^k\}$ satisfying (2). For the BS rule, it makes no difference which set of $\{f_i^k\}$ is used, because they all result in the same delay. For the BD rule, however, we want to find the particular set $\{f_i^{k*}\}$ results in maximum traffic bifurcation. Now for an adaptive routing rule, more traffic bifurcation means more traffic can be adaptively routed. And it has been shown [1] that as the adaptive portion of traffic increases relative to the fixed traffic, while keeping the total traffic constant, the delay performance improves. This is also true for the BD rule for the same reason.

To find $\{f_i^{k*}\}$, let us focus on a particular node, say node n. Let L_n be the set of outgoing links and define P_i^k as the probability of routing the kth commodity to link i at node n, or

$$P_i^k = \frac{f_i^k}{\sum_{j \in L_n} f_j^k} \qquad i \in L_n. \tag{4}$$

Recall that our objective is to find $\{f_i^{k*}\}$ such that the traffic bifurcation in the network is maximum. This is equivalent to finding the set of P_i^k's whose value are "as near to each other" as possible at each node. Therefore, we can find $\{f_i^{k*}\}$ by maximizing the traffic distribution entropy function[1] H:

$$H = -\sum_{k=1}^{M} \sum_{n=1}^{N} \left(\sum_{j \in L_n} f_j^k \right) \sum_{i \in L_n} P_i^k \log P_i^k$$

$$= -\sum_{k=1}^{M} \sum_{n=1}^{N} \sum_{i \in L_n} f_i^k \left[\log f_i^k - \log \left(\sum_{j \in L_n} f_j^k \right) \right] \tag{5}$$

subject to the nonnegativity constraints $f_i^k \geqslant 0$, the flow conservation constraints (3), and the link utilization constraints (2).

It can be shown that H is a linear combination of convex functions, and so is itself convex. The constraints are all linear. Therefore a unique maximum value of H exists and can be found by the same iterative technique prescribed in the last section. The flows $\{f_i\}$ and the *initial guessed values* of $\{f_i^k\}$ are obtained from the solution of the optimization problem in the preceding section. The set $\{P_i^{k*}\}$ which completely specifies the routing strategy is calculated from $\{f_i^{k*}\}$ at each node via (4).

B. Deterministic Routing Sequences

Instead of assigning routes randomly according to $\{P_i^{k*}\}$ at each node, we can use a deterministic routing sequence S_n^k:

$$S_n^k = \{s_1, s_2, \cdots, s_m\}$$

with $s_i = l$ meaning a routing decision to send the ith incoming messages of commodity k to outgoing link l at node n.

We now show how to calculate the decision sequence via a simple example. Consider the queueing system of Fig. 1(b), which is a model of the single node in Fig. 1(a). For the values of λ_1, λ_2, and λ shown, P_1 and P_2 are calculated to be 0.7 and

[1] Any change toward equalization of the probabilities $\{P_i^k\}$ increases H. Maximixing H therefore maximizes the amount of traffic bifurcation.

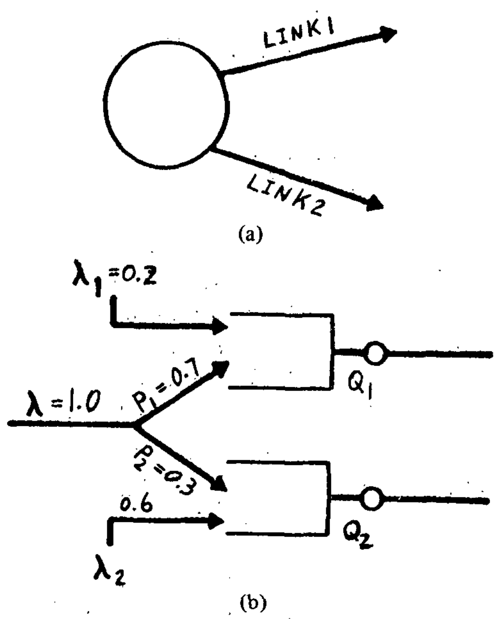


Fig. 1. (a) A network node with two outgoing links. (b) A two-queue system, model of the network node in (a).

0.3. We want to find a sequence of decisions, say $S = \{1\ 2\ 1\ 1\ 2\ \cdots\}$ such that the number of decisions of 1 (or on queue 1) and the number of decisions on 2 have ratios as close to 7:3 as possible for any segments of S. Compare this sequence to the randomly generated sequence S' where the probabilities of selecting "1" and "2" are 0.7 and 0.3, respectively. S appears to be more "orderly." Hence delay performance of the queueing system using S is improved when compared to that which uses S'. This we shall show in the next section.

For any subsequence of length m, let $D(1\,|m)$ be the number of 1-decisions and $D(2\,|m) = m - D(1\,|m)$ be the number of 2-decisions. We want $D(i\,|m)/m$, the fraction of messages to be routed to Q_i in a total of m message, be as close to P_i as possible for all m. Therefore, starting from $m = 1$, we choose the decision (1 or 2 in this case) that minimizes $\Sigma_{i=1}^{2} [D(i\,|m)/m - P_i]^2$ for each m. As an example, S for the λ messages of Fig. 1(b) is calculated to be $S = \{[1\ 2\ 1\ 1\ 1\ 2\ 1\ 1\ 2\ 1]\}$ where $[\cdot]$ *means that the sequence inside is to be repeated.* In general, for the set of P_i's, $i = 1, 2, \cdots, Q$, S is determined by the following algorithm.[2]

Step Q: $n = 1$

Step 1: Compute
$$e_i = \sum_{\substack{j=1 \\ j \neq i}}^{Q} \left(\frac{D(j\,|n-1)}{n} - P_j \right)^2 + \left(\frac{D(i\,|n-1)+1}{n} - P_i \right)^2$$
$\doteq$ squared error in making an i-decision at decision-node n.

Step 2: $e_k = \min\limits_{i} \left[\min\limits_{e_i} [e_1, e_2, e_3, \cdots] \right]$

$s_n = k$

Step 3: $n \leftarrow n + 1$; GO TO Step 1.

[2] To demonstrate the efficiency of this algorithm, the above sequence is calculated by hand.

In a network, we first calculate P_i^k from f_i^k. We then determine the sequences S_n^k from the P_i^k's for each commodity at each node. With the $\{S_n^k\}$ determined, the routing rule is completely specified.

As a final remark, we find out that the number of sequences we have generated are all recurrent. A closer study reveals the following theorem, which is needed in the analysis of the BD rule (Section IV). The proof is in the Appendix.

Theorem: The sequence of decisions S with rational P_i's given by $P_i = n_i/N$ is recurrent with period N, where $\{n_i\}$ and N are relatively prime.

C. An Example for Illustration

Consider the five-node network in Fig. 2(a). Let $\Gamma = [\gamma_{ij}]$ be the input traffic matrix with γ_{ij} the rate of external message arrival from node i to node j. All links are assumed to have a capacity of one and message lengths are also normalized to unity. For this simple network with symmetric input traffic and $\gamma_{23} = \gamma_{32} = 0$, it is sufficient to consider a unidirectional flow of traffic because the flow is the same in the reverse direction. If we focus on the flow from left to right, we have the queueing model and the rates of flow in and out of the queues as shown in Fig. 2(b). Using the BS rule for this simple network, it is easy to see that all messages would follow their unique minimum hop routes except the γ_{14} and the γ_{15} messages, which can be routed either through node 2 or node 3. How should these two message streams be bifurcated for minimum average time delay is what we are going to investigate. The f_i that minimizes (1) is given as $(f_1, f_2, f_3, f_4, f_5) = (0.8,\ 0.7,\ 0.6,\ 0.8,\ 0.8)$. Focusing on node 1 and its two outgoing links, we note that the four unknown flows due to individual commodities are $f_1^{1,4}$, $f_2^{1,4}$, $f_1^{1,5}$, and $f_2^{1,5}$. They are the flows due to $\gamma_{1,4}$ and $\gamma_{1,5}$ messages on links 1 and 2, respectively; and are related by the link utilization constraints (2) and the flow conservation constraints (3). Thus, from (2) we have

$$f_1 = f_1^{1,2} + f_1^{1,4} + f_1^{1,5} = 0.8$$

$$f_2 = f_2^{1,3} + f_2^{1,4} + f_2^{1,5} = 0.7.$$

Noting that $f_1^{1,2} = \gamma_{12} = 0.4$ and $f_2^{1,3} = \gamma_{13} = 0.4$, we arrive at

$$f_1^{1,4} + f_1^{1,5} = 0.4 \tag{6}$$

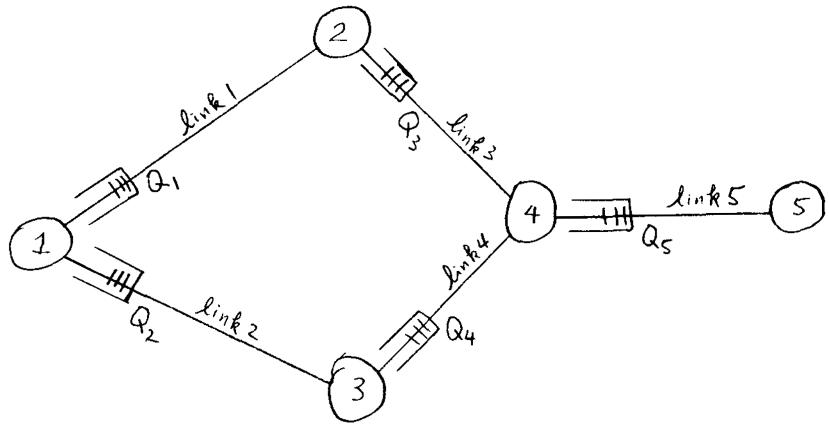
$$f_2^{1,4} + f_2^{1,5} = 0.3. \tag{7}$$

And from (3), we have

$$f_1^{1,4} + f_2^{1,4} = \gamma_{14} = 0.4 \tag{8}$$

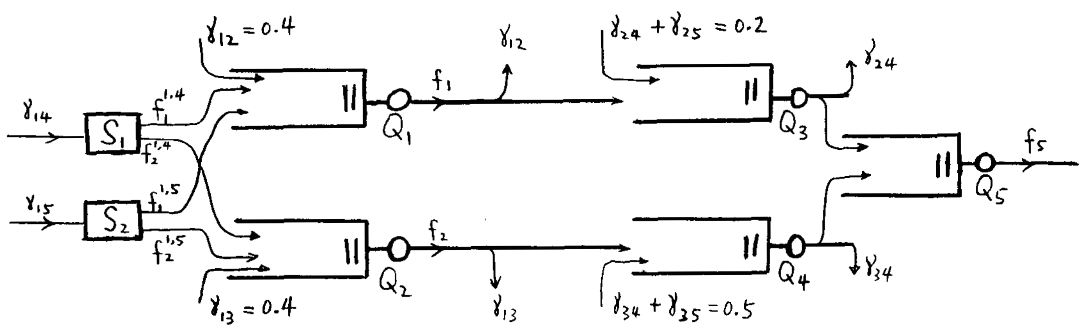
$$f_1^{1,5} + f_2^{1,5} = \gamma_{15} = 0.3. \tag{9}$$

Here are four equations for four unknowns. But, unfortunately, one of them is linearly dependent [(6) added to (7) is the same as (8) added to (9)]. So the $\{f_i^k\}$ cannot be uniquely determined and we have a little freedom in choosing the values of the individual commodity flows. As explained



$$\Gamma = [\gamma_{ij}] = \begin{bmatrix} - & 0.4 & 0.4 & 0.4 & 0.3 \\ 0.4 & - & 0 & 0.1 & 0.1 \\ 0.4 & 0 & - & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.3 & - & 0.2 \\ 0.3 & 0.1 & 0.2 & 0.2 & - \end{bmatrix}$$

(a)



(b)

Fig. 2. (a) The five-node network and the input traffic matrix. (b) The queueing model.

before, we want to choose the $\{f_i^k\}$ for maximum traffic bifurcation by maximizing H. For this example, we have

$$\begin{aligned} -H &= f_1^{1,4} \log f_1^{1,4} + f_2^{1,4} \log f_2^{1,4} + f_1^{1,5} \log f_1^{1,5} \\ &\quad + f_2^{1,5} \log f_2^{1,5} - (f_1^{1,4} + f_2^{1,4}) \\ &\quad \cdot \log (f_1^{1,4} + f_2^{1,4}) - (f_1^{1,5} + f_2^{1,5}) \\ &\quad \cdot \log (f_1^{1,5} + f_2^{1,5}) \\ &= f_1^{1,4} \log f_1^{1,4} + 2(0.4 - f_1^{1,4}) \log (0.4 - f_1^{1,4}) \\ &\quad + (f_1^{1,4} - 0.1) \log (f_1^{1,4} - 0.1) + 2.390. \end{aligned}$$

Differentiating H with respect to $f_1^{1,4}$, setting the result equal to zero and simplifying, we arrive at $f_1^{1,4} = 8/35$. Hence $f_2^{1,4} = 6/35$, $f_1^{1,5} = 12/70$, and $f_2^{1,5} = 9/70$. The γ_{14} traffic therefore is to be split into two streams with a 4:3 proportion for links 1 and 2; and for the γ_{15} traffic, also a 4:3 proportion. This therefore is the maximum bifurcation of traffic in node 1 while still preserving the flow rates on links 1 and 2 to be 0.8 and 0.7.

D. Network Operation

In the operation of the BD rule in a network under a varying traffic rate environment, it is assumed that a network control center (NCC) exists and recalculates the optimum flow distribution (in the sense of the BS rule criterion) whenever there are significant changes in the external flow pattern at the local nodes. It then determines the maximum traffic bifurcation of each commodity at each node by maximizing the H function. The patterns of traffic bifurcation (the P_i^k's) are sent to the respective nodes. This can be done either periodically or when necessary. Each node then generates the decision sequences from the P_i^k's received. The messages, after being classified by commodities (or their source-destination type), are then routed according to the decision sequences associated with their commodity.

We now come to discuss some practical considerations in implementing the BD rule. It should be noted that we are not documenting an existing rule. We are only pointing out the feasibility of this rule (or infeasibility, depending on the particular network being considered) based on some common technical requirements, such as traffic updates, computational overhead and traffic overhead.

1) Traffic rates usually vary during the hours of the day. Thus, they have to be updated periodically for efficient operation of the network. Under unusual circumstances such as node and link failure, however, immediate update is required. As an example, the input traffic to the CCIS network is telephone call initiated. The rates generally do not vary very much in any 15 minute intervals. In [14], the ARPA network HOST message arrival rate is shown on hourly basis. The update interval can be determined from these plots.

2) Computation of the optimum flow distribution is needed after each traffic update. There are two computational advantages.

i) The flow distribution of the last update is a suboptimal solution of the present update. Therefore, the time to reach the optimum solution is significantly less than the case where some arbitrary initial feasible solution is assumed.

ii) Due to random fluctuations, input traffic rates cannot be measured precisely. Therefore optimum flow distribution need not be highly accurate.[3] As an example, in [3] it was reported that the optimum flow computation time for the 21-node ARPA topology is 30 s (starting from an arbitrary initial feasible flow and with an accuracy of 10^{-4} on T, the overall average network delay). This is acceptable when the traffic update interval is in the order of minutes.

3) The local nodes need only send their estimates of the external arrival rates to the NCC when there is a significant change. On the other hand, a typical local node needs to receive only the f_i^k values that is i) associated with its outgoing links and ii) changed significantly from their previous values. Thus if the traffic update is not too frequent, the traffic overhead in the use of the BD rule is minimum.

IV. THE ANALYSIS OF THE BEST DETERMINISTIC RULE

In this section, we first analyze the BD rule in an isolated network node and study two degenerate cases. We then generalize the analysis to arbitrary networks and show that the BD rule always gives lower delay than the BS rule.

Consider an isolated network node with L outgoing links, all assumed to have the same capacity. Let λ_i represent the rate of message arrivals that is constrained to join queue $i(Q_i)$ and λ represent the rate of message arrivals that is constrained to join $Q_1, Q_2, \cdots, Q_L$ in proportion to P_1: P_2: P_3: $\cdots$ P_L (Fig. 3). Using the BD rule, let us first consider the state description of Q_1. Let N be the recurrence period of the routing sequence and $h(t)$, the position of the sequence at t. Then $h(t) = j$ with $s_j = m$ means that the next arrival of the λ message is to be routed to Q_m. Denote $q_1(t)$ as the number of

[3] As an example, if the traffic rate is measured to ±5 percent accuracy, there is no need to compute the flow distribution any more accurately than, say, two significant figures (1 percent error).

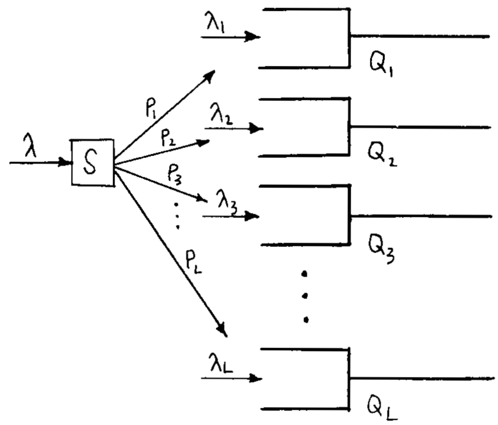


Fig. 3. An L-queue system with fixed arrivals to individual queues.

messages residing in Q_1 at t. The two quantities $q_1(t)$ and $h(t)$ completely specified Q_1 at t. Hence $[q_1(t), h(t)]$ is an appropriate state vector for a Markov process. The number of states for $h(t)$ is N and for $q_1(t)$ is $M + 1$, where M is the buffer capacity. The transition time between states has exponential distribution. Hence we can solve the steady state behavior of Q_1 by representing it as a two-dimensional Markov chain. Define

$$S_i'(j) = 1 \quad \text{when } s_j = i$$
$$= \quad \text{otherwise.}$$

Thus if

$$S = \{1\ 3\ 4\ 2\ 1\ 3\ 3\ 2\ 1 \cdots\},$$

we have

$$S_2 = \{0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0 \cdots\}.$$

Let $P_l(i, j) = \text{Prob}[q_l = i \text{ and } h = j]$. The state equations for Q_1 are

$$p_1(i,j) = \frac{1}{1 + \lambda + \lambda_1} [S_1'(j-1) \cdot \lambda \cdot P_1(i-1, j-1) + [1 - S_1'(j-1)] \cdot \lambda \cdot P_1(i, j-1) + \lambda_1 P_1(i-1, j) + P_1(i+1, j)] \qquad 1 \leq i \leq M-1 \qquad \forall_j \bmod N$$

$$P_1(0,j) = \frac{1}{\lambda + \lambda_1} [[1 - S_1'(j-1)] \cdot \lambda \cdot P_1(0, j-1) + P_1(1, j)] \qquad \forall_j \bmod N$$

$$P_1(M,j) = \frac{1}{1 + \lambda} [S_1'(j-1) \cdot \lambda \cdot P_1(M-1, j-1) + [1 - S_1'(j-1)] \lambda P_1(M, j-1) + \lambda_1 P_1(M-1, j)] \qquad \forall_j \bmod N.$$

The equations for other queues in Fig. 3 can be written down and solved independently. The occupancy probability for Q_l is

$$P_l(i) = \text{Prob } [i \text{ messages in } Q_l]$$
$$= \sum_{j=1}^{N} P_l(i,j) \qquad i = 1, 2, \cdots, M.$$

The expected length of Q_l, the overall delay, blocking probabilities for different traffic streams, etc. can all be calculated in the usual way [10].

To assess the delay performance of the BD rule, we consider two degenerate cases.

1) A two-queue system with arrival rate 2λ (Fig. 4). Assuming $1/\mu C = 1$, the utilization of each queue is $\rho = \lambda$. The BS rule assigns incoming traffic to Q_1 and Q_2 with equal probability. Using $M/M/1$ result, the average delay of messages is $D_S = 1/(\mu C - \lambda) = 1/(1 - \lambda)$. The BD rule routes arrival messages alternately to Q_1 and $Q_2 (S = \{[1\ 2]\})$. The interarrival time T of external arrivals is exponentially distributed with mean $1/2\lambda$. Since every other message joins Q_1, the interarrival time of those messages joining Q_1, denoted by T_1 is $T_1 = T + T$. Its density has transform

$$F_{T_1}(s) = \left[\frac{2\lambda}{s + 2\lambda}\right]^2.$$

This is the transform of the E_2 distribution. The BD rule therefore changed the Poisson statistic at the input end of the queue to the Erlangian statistic. By the technique in [9], the characteristic root of the above system is given by

$$\sigma = F_{T_1}(1 - \sigma).$$

The unique root in (0, 1) is $\sigma = (1 + 4\lambda - \sqrt{1 + 8\lambda})/2$. The average delay of messages in Q_1 is $D_D = 1/(1 - \sigma)$. (By symmetry, the messages in Q_2 have the same average delay.) To compare with the BS rule, we form the delay ratio

$$\frac{D_D}{D_S} = \frac{2(1 - \lambda)}{1 - 4\lambda + \sqrt{1 + 8\lambda}}.$$

For $\rho = \lambda = 0.5$, the ratio is 0.809. For $\rho \to 1$, the ratio approaches 0.75. The deterministic rule therefore gives 19-25 percent reduction of delay in the range $0.5 \leqslant \rho < 1$ when compared to the stochastic rule.

2) An L-queue system with arrival rate $L\lambda$ (Fig. 5). The BD rule degenerates to the sequential routing scheme with $S = \{[1\ 2\ 3 \cdots L]\}$. By symmetry, the queueing behavior is the same for all L queues. The arrival rate for any particular queue is $L\lambda/L = \lambda$. When L is large, the transform of interarrival time to Q_1 is

$$\lim_{L \to \infty} \left[\frac{\lambda}{s + L\lambda}\right]^L = e^{-s/\lambda}.$$

The interarrival time therefore, is a constant equal to $1/\lambda$

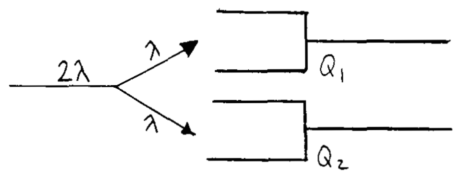


Fig. 4. A two-queue system with fixed arrivals suppressed.

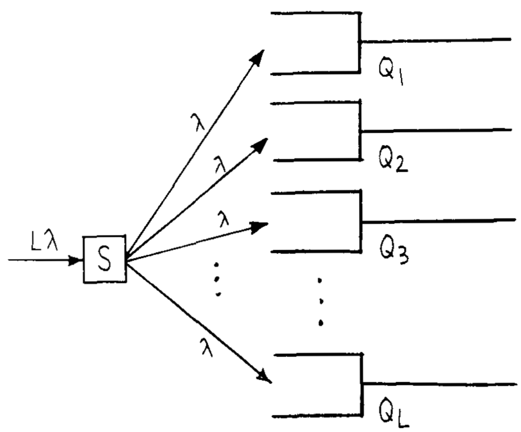


Fig. 5. An L-queue system with fixed traffic suppressed, $S = \{[1, 2, \cdots, L]\}$.

and Q_1 is essentially a $D|M|1$ queue (or equivalently an $E_\infty|M|1$ queue). The characteristic root is given by $\sigma = 1 + \lambda \ln \sigma$. Comparing the delay with that given by the BS rule, we have

$$\frac{D_D}{D_S} = \frac{1 - \sigma + \ln \sigma}{(1 - \sigma) \ln \sigma}.$$

For $\rho = 0.5$, $\sigma = 0.203$, the delay ratio is 0.627. As $\rho \to 1$, $\sigma \to 1$, the delay ratio approaches 0.5. We conclude that when the fixed traffic is suppressed, and with L large, the reduction of delay from the BS rule ranges from 37.3 percent to 50 percent in the range of $0.5 \leqslant \rho < 1$.

The analysis of the BD rule in a general network is similar to the join-biased-queue best stochastic (JBQ-BS) rule studied [1]. We first use the Poisson departure assumption.[4] This assumption says that for local routing rules (i.e., nonfeedback rules) with exponential messages, the departure process can be assumed as memoryless, or Poisson. It allows us to decouple queues at different nodes and analyze them separately. We have shown how to analyze queues at local nodes. The overall average time delay and the overall average blocking probability can be calculated from Little's formula applied to a network [10].

We can actually prove that the BD rule always gives lower delay than the BS rule. The argument is quite simple: since the utilization of each link is the same for both rules, we only need to compare the average length of each queue. For each queue, the above analysis (the delay ratio D_D/D_S) indicates that the BD rule (i.e., with Erlangian distributed interarrival time) always gives smaller average queue length than the BS

[4] This assumption has been used in [12], [13], [1] and possibly many other similar works in the analysis of queueing networks.

rule (i.e., with exponential distributed interarrival time). Since this is true for all queues (i.e., a network of Erlangian queues compared to a network of $M/M/1$ queues), we conclude that the BD rule always gives lower delay than the BS rule. Further, this argument is independent of the network size, the network topology and the input traffic assumed.

We refer the reader to [11] for an example of the application of the BD rule to the Common Channel Interoffice Signaling (CCIS) network for the telephone system.

V. SUMMARY AND CONCLUSION

We started our discussion with a brief view of the BS rule and continued on to make an extensive study of the BD rule. The BD rule was shown to always give lower delay than the BS rule and has the best delay performance among the fixed rules[5] in the literature. The operation of the BD rule in a varying traffic rate environment was described and its feasibility based on some common technical requirements was discussed.

APPENDIX
PROOF OF THE THEOREM IN SECTION III

Let us decompose the sequence generated into a sum of L subsequences: $S = S_1 + S_2 + \cdots S_L$ where L is the number of queues in the system. S_i has the property that all non-i-decisions are set to zero. Thus if

$$S = \{1\ 3\ 4\ 2\ 1\ 3\ 3\ 2\ 1\ \cdots\}$$

we have

$$S_2 = \{0\ 0\ 0\ 2\ 0\ 0\ 0\ 2\ 0\ \cdots\}.$$

S_i contains all the information needed for Q_i for routing.

Consider the kth decision of S_1, where $m\cdot N < k < (m+1)N$, m is any nonnegative integer. Suppose for the $(k-1)$ past decisions, l decisions are on 1 and $k-1-l$ decisions are on 0, or others. The decision rule for the kth decision will then be

$$\frac{mn_1 + (l+1)}{mN+k} - \frac{n_1}{N} \underset{0}{\overset{1}{\gtrless}} \frac{n_1}{N} - \frac{mn_1 + l}{mN+k}$$

by the given algorithm. Simplifying, we have

$$N[2mn_1 + 2l + 1] \underset{0}{\overset{1}{\gtrless}} 2n_1[mN+k] \quad \text{(A1)}$$

Now consider the $(k+N)$th decision; the decision rule is given as

$$\frac{2(m+1)n_1 + 2k + 1}{(m+1)N + k} \underset{0}{\overset{1}{\gtrless}} \frac{2n_1}{N}.$$

Upon simplifying, we have

$$N[2mn_1 + 2l + 1] \underset{0}{\overset{1}{\gtrless}} 2n_1[mN+k].$$

Comparing with (A1), we observe that the kth decision is the same as the $(k+N)$th decision. Since the above is true for arbitrary m, k, N, and n_1, we conclude that the sequence S_1 is recurrent with period N. We can go through the same argument for $Q_2, Q_3, \cdots$, etc., and establish that all S_i's are recurrent with period N. Now since S is the sum of all the S_i's, it is easy to see that S is also recurrent with the same period.

The case where equality holds in (A1) needs further explanation. This is the case where routing the incoming message to Q_1 or some other queues results in the same "error." Step 2 of the algorithm uniquely determined the queue to be selected. Since this determination depends only on the numbering of queues, the $(k+N)$th decision will be the same as the kth decision. Therefore, we conclude that blocks of N decisions located anywhere in S is recurrent.

[5] These are the rules which do not use the instantaneous queue length information in making routing decisions.

REFERENCES

[1] T. Yum, "Analysis of adaptive routing rule in computer communication networks," Ph.D. dissertation, Columbia Univ., 1978.

[2] J. McQuillan, "Adaptive routing algorithms for distributed computer networks," BBN, Inc. Rep., May 1974.

[3] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method," *Networks*, vol. 3, no. 3, 1973.

[4] T. Stern, "A class of decentralized routing algorithms using relaxation," *IEEE Trans. Commun.*, vol. COM-25, Oct. 1977.

[5] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Trans. Commun.*, vol. COM-25, Jan. 1977.

[6] L. Kleinrock, *Communication Nets.* New York: McGraw-Hill, 1964.

[7] D. G. Cantor and M. Gerla, "Optimal routing in a packet-switched computer network," *IEEE Trans. Comput.*, vol. C-23, Oct. 1974.

[8] M. Schwartz and C. K. Cheung, "The gradient projection algorithm for multiple routing in message switched networks," *IEEE Trans. Commun.*, vol. COM-24, Apr. 1976.

[9] L. Kleinrock, *Queueing Systems, Vol. 1: Theory.* New York: Wiley, 1975.

[10] M. Schwartz, *Computer-Communication Network, Design and Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1977.

[11] T. Yum, "A semidynamic deterministic routing rule in computer communication networks," in *Proc. Nat. Telecommun. Conf.*, Washington, DC, Nov. 1979.

[12] G. L. Fultz, "Adaptive routing techniques for message switching computer-communication networks," Univ. California, Los Angeles, Eng. Rep. UCLA-ENG-7252, July 1972.

[13] A. Livne, "Dynamic routing in computer-communication networks," Ph.D. dissertation, Dep. Elec. Eng., Polytechnic Inst. New York, 1977.

[14] L. Kleinrock, *Queueing Systems, Vol. 2, Computer Application.* New York: Wiley-Interscience, 1976, ch. 6, Fig. 6-26.

★

Tak-Shing P. Yum (S'76–M'78) received the B.S., M.S., Ph.D. degrees from Columbia University, New York, NY, in 1974, 1975, and 1978, respectively.

From 1978 to 1980, he was a member of the Technical Staff at Bell Laboratories, Holmdel, NJ, and worked in the area of data communications, mainly on dynamic routing, flow control, traffic engineering, and performance analysis of the Common Channel Interoffice Signaling Network. In September 1980, he joined the National Chiao-Tung University of Taiwan, where he is now Associate Professor in the Institute of Computer Engineering. His main research interest is in the design and analysis of computer-communication networks and in the performance analysis of computer systems.