# Resequencing of Messages in Communication Networks

TAK-SHING P. YUM, MEMBER, IEEE, AND TIN-YEE NGAI, STUDENT MEMBER, IEEE

*Abstract*—In this paper we study a message resequencing problem in a store-and-forward computer network where messages may go out of order while traversing logical channels. The logical channels are assumed to consist of multiple physical links which may be of different capacities. A message is dispatched to the fastest available link. Resequencing methods suggested in the literature [3] (resequencing at the channel level and resequencing at the virtual circuit level) are investigated for this link selection rule.

The analysis is done on a two-node network connected by multiple links. The source node together with the set of outgoing links are modeled as an $M/M/m$ queue with servers of different rates. The resequencing delay distribution and the average resequencing delay are derived.

On multihop networks, the effect of message length, link numbers, link service rates, and the resequencing methods on resequencing delay are investigated by simulation.

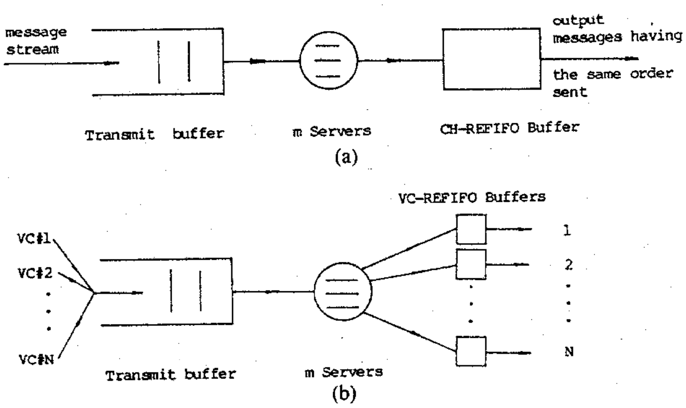



Fig. 1. (a) Channel level and (b) virtual circuit level resequencing.

## I. Introduction

Modern computer communication network architectures frequently use virtual circuits (VC's) to transmit messages across the network. Examples are the SNA (systems network architecture) proposed by IBM [1] and the French PTT's public network TRANSPAC [2]. For these networks, pairs of network nodes are connected by one or more logical channels consisting of multiple physical links.

For reasons of higher reliability and efficiency, messages on the same virtual circuit are often permitted to use different physical links while going over a logical channel. Consequently, variable length messages traversing through different capacity links may encounter different delays and transmission errors. This might cause messages to go out of sequence when they arrived at the destination node. Since most store-and-forward computer networks require messages to arrive at their destinations in the same order as they were generated, message resequencing protocols need to be implemented in these networks.

Recently, Bharath-Kumar and Kermani [3] analyzed such a resequencing problem in a two-node network connected by a single channel consisting of multiple *equal speed* links assuming that message arrivals form a Poisson process and message sizes are exponentially distributed. They proposed two resequencing methods. One is to resequence the messages at the channel level (CH-REFIFO). That is, all messages are assigned sequence numbers at the transmission side of the channel and the receive side reorders the messages using these numbers in one refifo box (or buffer). The second method is to resequence the messages at the virtual circuit level (VC-REFIFO). Here, messages belonging to the different VC's but which happen to be sharing the same multilink channel are reordered in the corresponding VC refifo boxes at the receiving end (Fig. 1). They also studied the effect of different capacity links and different message size mixes on the two resequencing methods by simulation.

In this paper, we generalize the CH-REFIFO and VC-REFIFO analysis of [3] on a single hop network to accommodate multiple links that may be of different bit rates. This generalization requires the specification of a message dispatching rule. Here, we assume that messages generated at the source node are queued in a channel output buffer of size $B$ units (each messages occupies one unit of buffer). They are then dispatched to the fastest idle link for transmission.

After the one-hop network analysis, we proceed to study the hop-by-hop resequencing strategy and the end-to-end resequencing strategy in a multihop network by simulation.

## II. Modeling and Analysis of a Single Hop Network

Consider a single outgoing logical channel consists of $m$ error-free links of possibly different speeds. Let the arrival of messages to the source node be a Poisson process with rate $\lambda$ and message lengths be exponentially distributed. It can then be modeled as an $M/M/m$ queue with servers of different rates as shown in Fig. 2. The transmission (service) time on link $j$ is then exponentially distributed with mean $u_j$, $j = 1, 2, \cdots, m$.

This $m$-server queue can be identified as an $m$-dimensional birth-and-death process. In [4] Cooper studied an $M/M/m$ queue with different service rates where customers are dispatched to the fastest server available. He derived the probability that an arriving customer will find all servers busy and the utilization of each server. We shall, however, introduce a recursive technique for solving numerically the set of state probabilities, and use the results to derive the channel and VC resequencing delay distributions. Resequencing delay is defined here as the time a message has to wait at the

Paper approved by the Editor for Network Protocols of the IEEE Communications Society. Manuscript received November 29, 1984; revised June 5, 1985. This paper was presented at IEEE INFOCOM '85, Washington, DC, March 1985.

T.-S. P. Yum is with the Department of Electronics, Chinese University of Hong Kong, N.T. Hong Kong.

T.-Y. Ngai is with International Quartz Limited, N.T. Hong Kong.

IEEE Log Number 8406874.

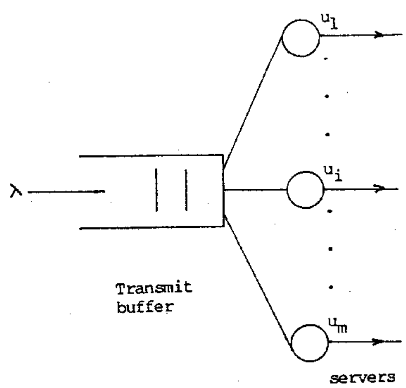



Fig. 2. $M/M/m$ system with servers of different rates.

destination node for all the messages it has bypassed during its transmission.

*A. State Probability Calculation*

Without loss of generality, we label the links such that $u_1 \geq u_2 \geq \cdots \geq u_m$. Let $X_j$ equal 1 or 0 to denote whether link $j$ is busy or idle. Also let $Q$ be the number of customers waiting in the queue, $(X; Q)$ be the state vector, $P(x; q) = \text{Prob}\ [X_1 = x_1, \cdots, X_m = x_m;\ Q = q]$ be the equilibrium state probability, and $B$ be the transmit buffer size. For simplicity of notations, let

$$\mathbf{1} = (1, 1, \cdots, 1)$$

$$\mathbf{1}_j = (0, \cdots, 0, \overset{j\text{th}}{1}, 0, \cdots, 0)$$

and

$$\delta(x, y) = \begin{cases} 0 & \text{when } x \neq y \\ 1 & \text{when } x = y. \end{cases}$$

The equilibrium probabilities then satisfy the following sets of flow equations.

i) For $q = 0$ and $x \neq \mathbf{1}$:

$$\left(\lambda + \sum_{j=1}^{m} u_j x_j\right) P(x; 0) = \lambda \sum_{j=1}^{m} \delta\left(j, \sum_{i=1}^{j} x_i\right) P(x - \mathbf{1}_j; 0) + \sum_{j=1}^{m} u_j \delta(0, x_j) P(x + \mathbf{1}_j; 0). \tag{1}$$

ii) For $0 \leq q \leq B - 1$ and $x = \mathbf{1}$:

$$\lambda P(\mathbf{1}; q) = (u_1 + u_2 + \cdots + u_m) P(\mathbf{1}; q+1), \tag{2}$$

let $u_s = u_1 + u_2 + \cdots + u_m$ and $\rho \triangleq \lambda / u_s$. Then

$$P(\mathbf{1}; q) = \rho^q P(\mathbf{1}; 0). \tag{3}$$

For $B$ finite, the solution always exists. For $B$ being infinite, $\rho$ is required to be less than 1.

Let $\mathbf{I}_r$ denote the $2^r \times 2^r$ identity matrix, and define the set of square matrices $\mathbf{A}_0, \mathbf{A}_1, \cdots, \mathbf{A}_{m-1}$ recursively as follows:

$$\mathbf{A}_0 = \lambda,$$

$$\mathbf{A}_1 = \begin{pmatrix} \lambda & -u_1 \\ -\lambda & \lambda + u_1 \end{pmatrix},$$

$$\mathbf{A}_2 = \left[\begin{array}{cc:cc} \lambda & -u_1 & -u_2 & 0 \\ -\lambda & \lambda + u_1 & 0 & -u_2 \\ \hdashline 0 & 0 & \lambda + u_2 & -u_1 \\ 0 & -\lambda & -\lambda & \lambda + u_1 + u_2 \end{array}\right] = \left[\begin{array}{c:c} \mathbf{A}_1 & -u_2 \mathbf{I}_1 \\ \hdashline \begin{matrix} 0 & 0 \\ 0 & -\lambda \end{matrix} & \mathbf{A}_1 + u_2 \mathbf{I}_1 \end{array}\right].$$

Therefore, in general, $\mathbf{A}_r$ is

$$\mathbf{A}_r = \left[\begin{array}{c:c} \mathbf{A}_{r-1} & -u_r \mathbf{I}_{r-1} \\ \hdashline \begin{matrix} 0 & 0 \\ 0 & -\lambda \end{matrix} & \mathbf{A}_{r-1} + u_r \mathbf{I}_{r-1} \end{array}\right]_{2^r \times 2^r} \qquad r = 1, 2, \cdots, m-1. \tag{4}$$

Let $\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_m$ be another set of matrices defined as

$$\mathbf{D}_r = \mathbf{A}_{r-1} + \left(\sum_{i=r+1}^{m} u_i\right) \mathbf{I}_{r-1} \qquad r = 1, 2, \cdots, m. \tag{5}$$

For simplicity, denote $P(x; 0)$ as $P(x)$. Then (5) can be written in the following matrix form with $q(\cdot) = P(\cdot)/P(\mathbf{1})$:[1]

$$\left[\begin{array}{cccccc} \mathbf{D}_m & u_m \mathbf{I}_{m-1} & & & & \\ & \mathbf{D}_{m-1} & u_{m-1} \mathbf{I}_{m-2} & & & \\ & & \ddots & & & \\ & \bigcirc & & \mathbf{D}_r & u_r \mathbf{I}_{r-1} & \\ & & & & \ddots & \\ & & & & \mathbf{D}_1 & u_1 \\ & & & & 0 & 1 \end{array}\right] \cdot \begin{bmatrix} q(0, \cdots, 0) \\ \vdots \\ q(1, \cdots, 1, 0) \\ q(0, \cdots, 0, 1) \\ \vdots \\ q(0, 1, \cdots, 1) \\ q(1, \cdots, 1) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{6}$$

Using sparse matrix technique, this set of equations can be solved recursively starting from $r = 1$ as follows:[2]

[1] In other words, we first seek the relative values of $P(\cdot)$'s to $P(\mathbf{1})$ and then use the property that all probabilities sum to 1 to find their absolute values in the second stage.

[2] For an $m$-server system, the time complexity of evaluating (6) using the brute force method is $O(2^{2m})$. However, if recursive method is used, the time complexity is $O(2^{2m-2})$. The memory space requirement is reduced by half.

$$D_r\begin{pmatrix} q(\overbrace{0, \cdots, 0}^{r}, 1, \cdots, 1) \\ \vdots \\ q(1, \cdots, 1, 0, 1, \cdots, 1) \end{pmatrix} = u_r\begin{pmatrix} q(\overbrace{0, \cdots, 0}^{r-1}, 1, \cdots, 1) \\ \vdots \\ q(1, \cdots, 1, 1, \cdots, 1) \end{pmatrix} \qquad r=1, 2, \cdots, m. \tag{7}$$

As an example, for $m = 4$,

$$\mathbf{D}_1 q(0, 1, 1, 1) = (\lambda + u_2 + u_3 + u_4) q(0, 1, 1, 1) = u_1 q(1, 1, 1, 1)$$

$$\mathbf{D}_2\begin{pmatrix} q(0, 0, 1, 1) \\ q(1, 0, 1, 1) \end{pmatrix} = \begin{pmatrix} \lambda + u_3 + u_4 & -u_1 \\ -\lambda & \lambda + u_2 + u_3 + u_4 \end{pmatrix}\begin{pmatrix} q(0, 0, 1, 1) \\ q(1, 0, 1, 1) \end{pmatrix} = u_2\begin{pmatrix} q(0, 1, 1, 1) \\ q(1, 1, 1, 1) \end{pmatrix}.$$

We can now use the fact that all probabilities sum to 1 to obtain the value of $P(\mathbf{1})$. Let us assume that the transmit buffer is of size $B$ and all overflow messages are rejected. Adding all probabilities, we have

$$\sum_{\substack{\text{all}\\ \text{states}}} P(x; q) = \sum_{x \neq 1} P(x) + \sum_{q=0}^{B} P(\mathbf{1}; q) = \sum_{x \neq 1} q(x) \cdot P(\mathbf{1}) + \frac{1-\rho^{B+1}}{1-\rho} P(\mathbf{1}) = 1. \tag{8}$$

Therefore,

$$\frac{1}{P(\mathbf{1})} = \sum_{x \neq 1} q(x) + \frac{1-\rho^{B+1}}{1-\rho}. \tag{9}$$

The average queue length is then

$$L_q = \sum_{q=1}^{B} qP(\mathbf{1}; q) = P(\mathbf{1})\rho\left(\frac{1-(B+1)\rho^B + B\rho^{B+1}}{(1-\rho)^2}\right). \tag{10}$$

The average queueing time is

$$t_q = \frac{L_q}{\lambda(1-PB)} \tag{11}$$

where $PB = P(\mathbf{1}; B)$ is the message loss probability due to buffer overflow. Similarly, the average service time using Little's result is

$$t_s = \frac{L_s}{\lambda(1-PB)} \tag{12}$$

where

$$L_s = \sum_{x} \|x\| P(x; 0) + M \sum_{q=1}^{B} P(\mathbf{1}; q)$$

and

$$\|x\| = \sum_{i=1}^{m} x_i.$$

For the limiting case of $B \to \infty$, we have

$$\frac{1}{P(\mathbf{1})} = \frac{1}{1-\rho} q(x), \quad L_q = P(\mathbf{1}) \frac{\rho}{(1-\rho)^2}$$

and $PB = 0$.

### B. CH-Refifo Delay Distribution

Let a tagged message which is about to begin transmission see the set of links in condition $x$. Also let $S = \{i | X_i = 1\}$ and $I = \{j | X_j = 0\}$ be the set of busy and idle links, respectively. The tagged message is assigned to the lowest indexed idle link, or link $k$ where $k = \min [j | j \in I]$.

Let us assume that the tagged message bypasses the messages in the links indexed by a set $U$, which is a subset of $S$. Let the transmission time of the tagged message in link $k$ be $t$. Due to the memoryless property of transmission times, the probability that a message is still in transmission in link $j$ after an elapsed time $t$ is $e^{-u_j t}$. Since the message transmission times are all mutually independent random variables, the probability that the tagged message exactly bypasses the messages in $U$ after an elapsed time $t$ is

$$P_{bp}(U | t, S) = \left(\prod_{j \in U} e^{-u_j t}\right)\left(\prod_{j \in S-U} (1-e^{-u_j t})\right). \tag{13}$$

Removing the condition on $t$, the probability that the tagged message bypasses exactly the messages in $U$, given that the links in $S$ are busy, is

$$\begin{aligned} P_{bp}(U|S) &= \int_{t=0}^{\infty} P_{bp}(U|t, S)\, dF_{T_k}(t) \\ &= \int_0^{\infty} \left(\prod_{j \in U} e^{-u_j t}\right)\left(\prod_{j \in S-U} (1-e^{-u_j t})\right) u_k e^{-u_k t}\, dt \\ &= \frac{u_k}{u_k + \sum_{j \in U} u_j} \sum_{i \in S-U} \left[\frac{u_i}{u_k + u_i + \sum_{j \in U} u_j} \cdot \sum_{l \in S-U-\{i\}} \left[\frac{u_l}{u_k + u_i + u_l + \sum_{j \in U} u_j} \cdots \sum_{\substack{r \in S-U-\{i\} \\ -\{l\}}} \frac{u_r}{u_k + \sum_{j \in S} u_j}\right] \cdots\right] \end{aligned} \tag{14}$$

where $F_{T_k}(t)$ is the transmission time distribution function of the tagged message in link $k$.

Let $W_j$ be the waiting time of the tagged message given that it bypassed the message in link $j$. Then its distribution function is

$$F_{W_j}(y) = 1 - e^{-u_j t} \tag{15}$$

since all messages are exponentially distributed.

The waiting time of the tagged message given that it bypasses the messages in $U$ is

$$W_{U,S} = \max_{j \in U} [W_j]. \tag{16}$$

Since the $W_j$'s are mutually independent,

$$\text{Prob } [W_{U,S} \le t] = \prod_{j \in U} (1 - e^{-u_j t}). \quad (17)$$

Removing the condition on $U$,

$$\text{Prob } [W_S \le t] = \sum_{U \in U_S} \prod_{j \in U} (1 - e^{-u_j t}) P_{bp}(U|S) \quad (18)$$

where $U_S$ is the set of all subsets of $S$, including $S$ and the empty set, and $P_{bp}(U|S)$ is given by (14).

Finally, removing the condition on $S$, the waiting time distribution function is

$$F_W(t) = \sum_{S \in U_L} \text{Prob } [W_S \le t] Q[S] \quad (19)$$

where $U_L$ is the set of all proper subsets of $L = \{1, 2, \cdots, m\}$, i.e., excluding $L$ but including the empty set;[3] and $Q[S]$ is the probability that the tagged message joins the other messages in $S$, and is related to the state probabilities as follows:

$$Q[S] = \begin{cases} P(x; 0) & \text{for } \|S\| < m-1, \\ & \text{i.e., more than one link idling;} \\ \left(\sum_{q=0}^{B} P(1; q)\right)\left(\dfrac{u_i}{\sum_{j=1}^{m} u_j}\right) + P(x; 0) & \\ & \text{for } \|S\| = m-1 (S = L - \{i\}), \\ & \text{i.e., only link } i \text{ is idle;} \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

where $\|A\|$ denotes the number of elements in set $A$.

Using the moment generation function, the average refifo delay $\bar{W}$ is found as

$$\bar{W} = \sum_{S \in U_L} Q[S] \sum_{U \in U_S} P_{bp}(U|S) \bar{W}_{U,S} \quad (21)$$

where

$$\bar{W}_{U,S} = \int_0^\infty \left[1 - \prod_{j \in U} (1 - e^{-u_j t})\right] dt$$
$$= \left(\sum_{j \in U} \frac{1}{u_j}\right) - \left(\sum_{\substack{j,k \in U \\ j \ne k}} \frac{1}{u_j + u_k}\right) + \cdots + (-1)^{\|U\|+1} \left(\frac{1}{\sum_{j \in U} u_j}\right) \quad (22)$$

is the conditional average waiting time.

Similarly, the second moment of the refifo time is obtained as

$$E[W^2] = \sum_{S \in U_L} Q[S] \sum_{U \in U_S} P_{bp}(U|S) E[W^2(U, S)] \quad (23)$$

where

$$E[W^2(U, S)] = \left(\sum_{j \in U} \frac{1}{u_j^2}\right) - \left(\sum_{j,k \in U} \frac{1}{(u_j + u_k)^2}\right) + \cdots + (-1)^{\|U\|+1} \left(\frac{1}{\left(\sum_{j \in U} u_j\right)^2}\right) \quad (24)$$

upon integrating over the distribution function.

At equilibrium, the mean input rate to the refifo box equals $\lambda(1 - PB)$. Hence, the average queue size of the refifo buffer is $L_R = \bar{W}\lambda(1 - PB)$ by Little's formula.

### C. Analysis of VC-REFIFO

Let $N$ be the total number of virtual circuits (VC). Assuming that the messages for a particular VC are generated from a Poisson source and let $\lambda_i$ be the mean arrival rate from the $i$th virtual circuit (VC $i$); then $\lambda = \sum_{i=1}^{N} \lambda_i$ is the total arrival rate to the channel.

The messages in each VC are resequenced separately in each VC refifo box. Let

$$P_{VCi}[U|S] = \text{Prob [a VC } i \text{ message bypasses a set of VC } i \text{ messages in } U|S]$$
$$= \sum_{\substack{V \in U_S \\ U \subseteq V}} \text{Prob [a VC } i \text{ message bypasses the messages in } V \text{ and the messages in the subset } U \text{ of } V \text{ are in VC } i|S]$$
$$= \sum_{\substack{V \in U_S \\ U \subseteq V}} \text{Prob [a VC } i \text{ message bypasses the messages in } V|S] \times \text{Prob [the messages in the subset } U \text{ of } V \text{ are in VC } i]$$
$$= \sum_{\substack{V \in U_S \\ U \subseteq V}} P_{bp}(V|S) \alpha_i^{\|U\|} (1 - \alpha_i)^{\|V - U\|} \quad (25)$$

where $P_{bp}(V|S)$ is given by (14) and $\alpha_i = \lambda_i/\lambda$ is the probability that a message in a link is from VC $i$.

Following the same procedures in the CH-REFIFO analysis, the refifo delay distribution for VC $i$ can be obtained from (19), from which the average delay $\bar{W}_i$ and variance $(\sigma_{W_i})^2$ can also be found. The average occupancy of the $i$th resequencing buffer $L_R(i)$ is

$$L_R(i) = \bar{W}_i \lambda_i (1 - PB). \quad (26)$$

## III. Comparisons of Resequencing Methods and the Design of Refifo Buffer

Consider a three-link system with $u_1 = 3$, $u_2 = 2$, and $u_3 = 1$. In Fig. 3, we show the average refifo delay for the CH-REFIFO and the VC-REFIFO strategies. Two virtual circuits are assumed in the VC-REFIFO case. We see that the refifo delay increases *gradually* with system utilization.

Since the refifo delay constitutes only part of the total time in system, the performance of the system should be measured by the total delay, i.e., the sum of the queueing, the transmission, and the refifo delays. In Fig. 4, curve *a* shows the delay for the one-link system with the total combined capacity, i.e., $u = u_1 + u_2 + u_3 = 6$. There, the resequencing problem is avoided altogether but the system lacks the reliability property of a multilink diversity system. Curve *b* shows the total delay for CH-REFIFO versus the utilization for the same three-link system. Curve *c* shows the delay performance of the same three-link system. But here, in

[3] Note that the total of elements in $U_L$ is $2^m$ and is a large number for a moderate size $m$ value. The most computationally taxing part of this analysis, however, is the evaluation of $P(x, 0)$. In the examples given, we have limited $m \le 3$ although $m = 4$ or 5 is also feasible. This analysis, however, is general.

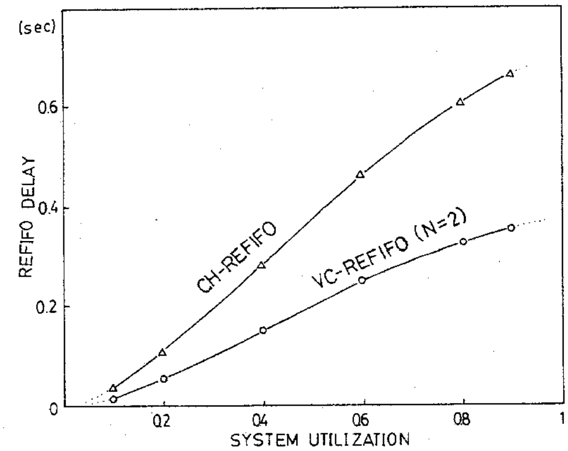



Fig. 3. Refifo delay versus system utilization: three-link system with infinite buffer, $u_1 = 3$ messages/s, $u_2 = 2$ messages/s, and $u_3 = 1$ message/s.

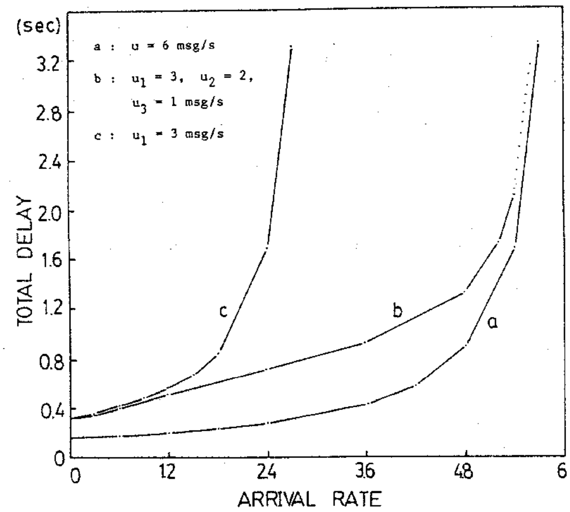



Fig. 4. Total delay versus arrival rate: three-link system with $u_1 = 3$ messages/s, $u_2 = 2$ messages/s, and $u_3 = 1$ message/s. Infinite buffer.

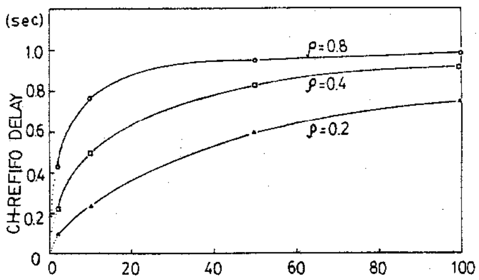



Fig. 5. CH-REFIFO delay versus link service rate mismatch: two-link system with $u_2 = 1$ message/s and $u_1$ varies from 2 to 100 messages/s.

order to eliminate the resequencing problem, the system always use the highest speed link, i.e., link 1 with $u_1 = 3$. Comparing curves $b$ and $c$, we see that to minimize the total delay, it is better to use all three links.

Fig. 5 illustrates the effect of the service rate difference on the refifo delay for the two-link system with $u_1 = 1$ and $u_2$ varies from 2 to 100. We see that the higher the difference of service rates, the longer the refifo delay. Hence, to minimize refifo delay, link speeds should be chosen as evenly as possible.

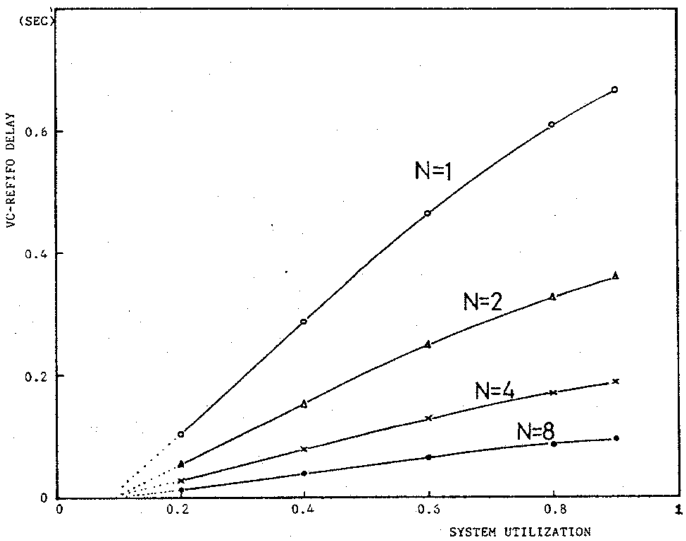



Fig. 6. VC-REFIFO delay versus system utilization: three-link system with $u_1 = 3$ messages/s, $u_2 = 2$ messages/s, and $u_3 = 1$ message/s. Infinite buffer.

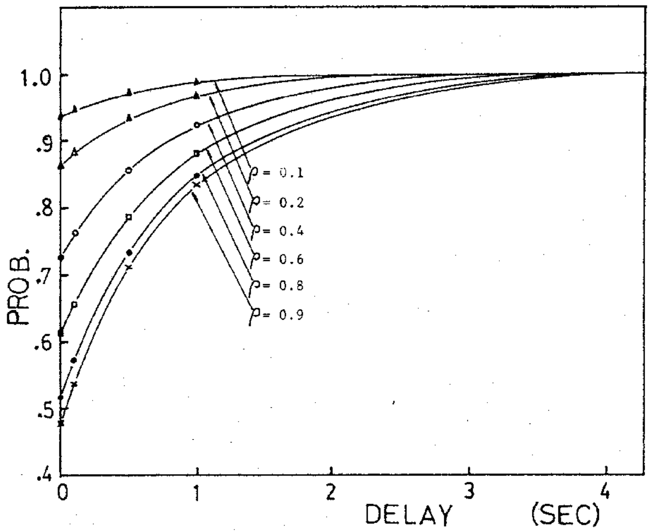



Fig. 7. CH-REFIFO delay distribution: two-link system with $u_1 = 2$ messages/s, $u_2 = 1$ message/s. Infinite buffer.

Fig. 6 shows the VC-REFIFO delay on a three-link system with $N$, the number of virtual circuits, as a parameter. Each virtual circuit is assumed to carry the same load. The VC-REFIFO strategy with only one VC degenerates to the CH-REFIFO. We find that the VC-REFIFO delay decreases with increasing virtual circuit number $N$. In fact, for a two-link system, the $N$ VC VC-REFIFO delay is exactly $1/N$ of the CH-REFIFO delay. This can easily be proved by direct substitution of equations derived.

The VC-REFIFO and CH-REFIFO delay distributions of a two-link system are given in Figs. 7 and 8, respectively. We see that the probability of no resequencing delay is quite large. Even at $\rho = 0.9$, the probability is 0.48 for CH-REFIFO and 0.74 for VC-REFIFO with $N = 2$. We also see that for CH-REFIFO, the probability that a message has to wait longer than 3.5 s is practically zero for all system utilization factors. Hence, the maximum refifo buffer required should be about $3.5(u_1 + u_2) = 10$ message units. The average refifo buffer occupancy, however, is calculated to be only 1.26. For VC-

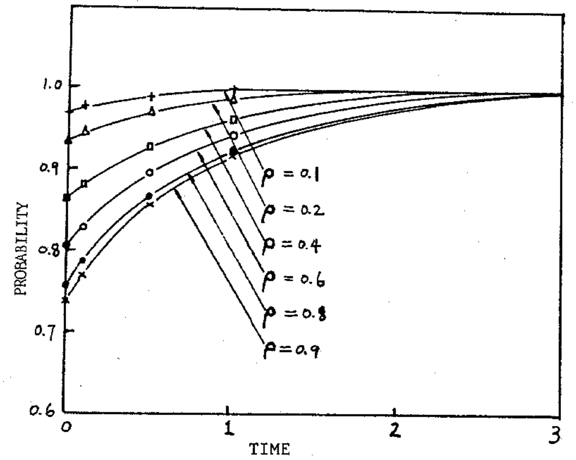



Fig. 8. VC-REFIFO delay distribution: two-link system with $N = 2$, $u_1 = 2$ messages/s, and $u_2 = 1$ message/s. Infinite buffer.

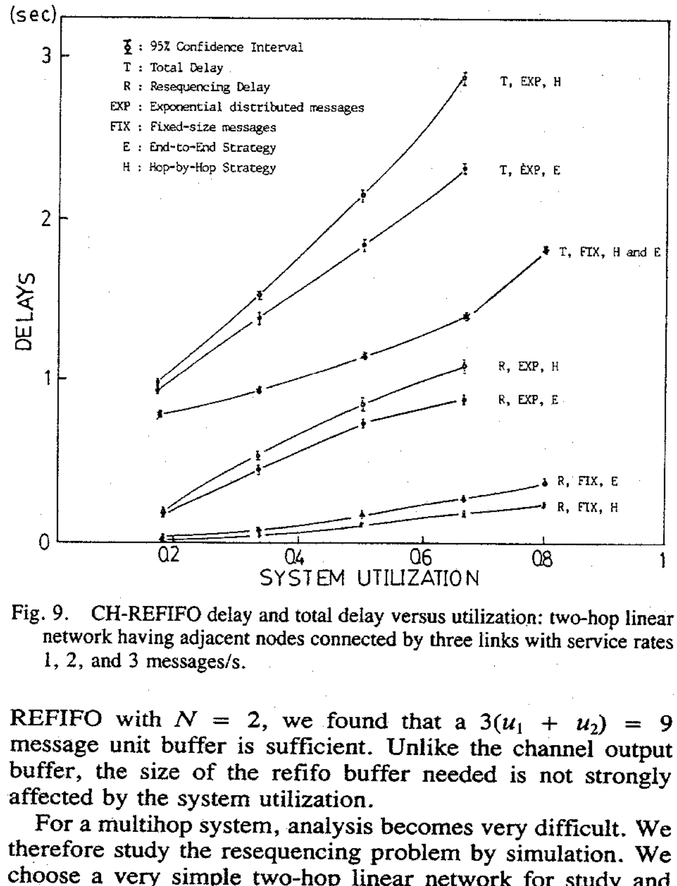



Fig. 9. CH-REFIFO delay and total delay versus utilization: two-hop linear network having adjacent nodes connected by three links with service rates 1, 2, and 3 messages/s.

REFIFO with $N = 2$, we found that a $3(u_1 + u_2) = 9$ message unit buffer is sufficient. Unlike the channel output buffer, the size of the refifo buffer needed is not strongly affected by the system utilization.

For a multihop system, analysis becomes very difficult. We therefore study the resequencing problem by simulation. We choose a very simple two-hop linear network for study and investigate two kind of resequencing strategies: hop-by-hop resequencing where messages are resequenced before forward to the next node, and end-to-end resequencing where messages are resequenced only at the destination node. Both exponential and fixed size messages are considered and adjacent nodes are connected by three links with service rates 1, 2, and 3 messages/s, respectively.

Fig. 9 shows the CH-REFIFO delays and total delays on the two-hop linear network for both exponential and fixed-size messages. We see that for fixed size messages, the hop-by-hop resequencing strategy gives smaller average resequencing delay compared to the end-to-end strategy. But interestingly, there is no significant difference in their total average delays.

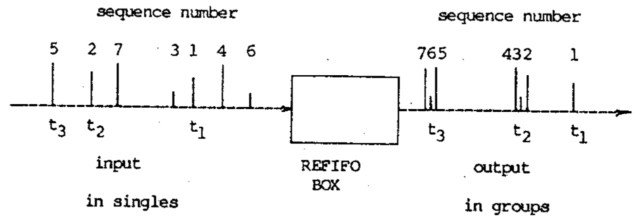



Fig. 10. A snapshot on the input and output processes of a refifo box.

In contrast, for exponentially distributed messages, the end-to-end resequencing strategy gives smaller average resequencing delay as well as total delay. We have no satisfactory explanation for this difference due to message size distribution. To have a better understanding of the resequencing phenomena, more extensive simulation is needed on a general network and with interfering traffic.

## V. Conclusions and Suggestions for Further Research

In this paper, we derived the channel level resequencing and the VC level resequencing delay distributions of a one-hop network with multiple links. We found that CH-REFIFO is equivalent to VC-REFIFO with one VC, and that if the input load is equally divided by $N$ virtual circuits, the VC-REFIFO delay will be approximately $1/N$ the CH-REFIFO delay.

Simulation results show that the hop-by-hop resequencing strategy gives better performance than the end-to-end resequencing strategy on a three-hop linear network with exponential messages. Simulation of a more general network is needed to establish the general validity of the conclusion drawn.

For further investigation, the assumptions of error-free transmission links and infinite refifo buffer can be dropped. Messages then need to be retransmitted when they are discarded due to transmission error or resequencing buffer overflow. The retransmitted messages are more likely to be bypassed by other messages. Hence, the resequencing time will, in general, be increased. The effect of different acknowledgment schemes [9] on refifo delay also need to be studied. Flow control mechanisms [10], [11] also affect the resequencing time and further analysis is needed. Some savings of buffer space could be achieved by combining the resequencing buffer and the transmit buffer into one common buffer in a node. This, however, will be at the expense of a more complex buffer management scheme. Detail analysis is needed for this shared buffer organization.

We observed that the output process of the refifo box has a very large coefficient of variation compared to the input process. Characterization of this process is needed to extend the resequencing analysis to a multihop network. A snapshot of the input and the output processes is shown in Fig. 10.

Another approach to the refifo problem is to use the job-shop scheduling method. The links can be modeled as processors of different speeds while the messages are tasks in sequence. The general job-shop scheduling problem is to maximize the utilization under the constraint of some precedent rule on the tasks [12]. However, in the refifo problem, no precedency of the tasks is set. Instead, it requires the scheduled tasks to be finished in a certain sequence. Furthermore, since the input process is random, dynamic scheduling is required for each incoming task. This appears to be a difficult problem, and seeking suboptimal solutions by heuristic algorithms may be the only feasible approach. The design and analysis of such heuristic algorithms is also a challenging topic.

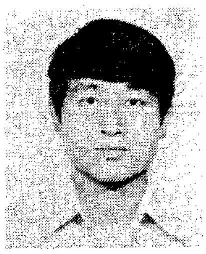
ACKNOWLEDGMENT

Suggestions made by the three anonymous referees were of considerable help to us in improving the quality of this paper. M. W. Chan has helped us to obtain some simulation results and has also drawn a number of figures for us.

REFERENCES

[1] A. S. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1981, pp. 187–196.
[2] A. Danet, R. Despres, A. LeRest, G. Pichon, and S. Ritzenthaler, "The French public packet switching service: The TRANSPAC network," in *Proc. 3rd Int. Comput. Commun. Conf.*, 1976, pp. 251–260.
[3] K. Bharath-Kumar and P. Kermani, "Analysis of a resequencing problem in communication networks," in *Proc. IEEE INFOCOM,* Apr. 1983, pp. 303–310.
[4] R. B. Cooper, "Queues with ordered servers that work at different rates," *Opsearch,* vol. 13, no. 2, pp. 69–77, 1976.
[5] T.-Y. Ngai, "Resequencing of messages in communication networks," Master's thesis, Dep. Electron., Chinese Univ. Hong Kong, June 1984.
[6] R. B. Cooper, *Introduction to Queueing Theory,* 2nd ed. London, England: Arnold, 1981, ch. 4, pp. 123–175.
[7] D. Fakinos, "The *M/G/k* blocking system with heterogeneous servers," *J. Oper. Res. Soc.*, vol. 31, pp. 919–927, 1980.
[8] J. S. Kaufman, "Congestion formulas for a heterogeneous server loss system with random selection discipline," *Oper. Res.*, vol. 29, no. 6, pp. 1167–1180, 1980.
[9] S. S. Lam, "Store-and-forward buffer requirements in a packet switching network," *IEEE Trans. Commun.*, vol. COM-24, pp. 394–403, 1976.
[10] M. Schwartz, *Computer-Communication Network Design Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1977, ch. 4, pp. 212–261.
[11] M. Gerla and L. Kleinrock, "Flow control: A comparative study," *IEEE Trans. Commun.*, vol. COM-28, pp. 553–574, 1980.
[12] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*. Reading, MA: Addison-Wesley, 1967.

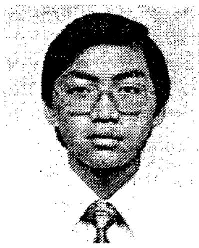
★



**Tak-Shing P. Yum** (S'76–M'78) was born in Shanghai, China, on February 26, 1953. He received the B.S., M.S., and 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 mainly on the performance analysis of the common channel interoffice signaling network. From September 1980 to June 1982, he was an Associate Professor at the Institute of Computer Engineering, National Chiao-Tung University, Taiwan. Since August 1982, he has been a Lecturer in the Department of Electronics, Chinese University of Hong Kong. His main research interest is in the design and analysis of computer communication networks.

★



**Tin-Yee Ngai** (S'78) received the B.Sc. degree in electronics and the M. Phil. degree from the Chinese University of Hong Kong in 1982 and 1984, respectively.

He is now an Engineer at International Quartz Limited, Hong Kong. His research interests are in communication networks, operational research, and management science.