(2)

Naval Research Laboratory

Washington, DC 20375-5000

AD-A243 584

DTIC
ELECTE
DEC 20 1991
S C D

NRL Report 9359

A Study of Channel-Access Schemes for Integrated Voice/Data Radio Networks

JEFFREY E. WIESELTHIER

Communication Systems Branch
Information Technology Division

ANTHONY EPHREMIDES

Electrical Engineering Department
University of Maryland
College Park, Maryland 20742

and

Locus, Inc.
Alexandria, Virginia 22303

November 29, 1991

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 29, 1991	Interim 7/89 to 6/91

4. TITLE AND SUBTITLE

A Study of Channel-Access Schemes for Integrated Voice/Data Radio Networks

5. FUNDING NUMBERS

PE - 61153N
PR - RR021-0542
WU - DN159-036

6. AUTHOR(S)

Jeffrey E. Wieselthier and Anthony Ephremides*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL Report 9359

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

A. Ephremides is with the University of Maryland and Locus, Inc.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this report we address the major issues associated with channel access in integrated wireless networks, and we propose and analyze the Pure-Reservation Voice/Data Non-Interleaved-Frame Fixed-Length (PR-VD-NIFFL) protocol. This scheme is well suited to either satellite or to terrestrial networks. A two-dimensional first-order Markov chain model for this scheme is presented, and techniques that exploit the structural properties of this chain to simplify the evaluation of the equilibrium state, without sacrificing accuracy, are described. Analytical models for the evaluation of data-packet delay for both fixed- and movable-boundary versions of this protocol and for voice-call blocking probability are presented. Computational results illustrate the dependence of performance on system parameters and demonstrate the improvement that can be achieved through the use of the movable-boundary version.

14. SUBJECT TERMS

Communications network | Voice/data integration
Multiple access

15. NUMBER OF PAGES

45

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

CONTENTS

A STUDY OF CHANNEL-ACCESS SCHEMES FOR INTEGRATED VOICE/DATA RADIO NETWORKS

1. INTRODUCTION

Channel-access methods for integrated voice/data radio networks should reflect the different requirements on delay and error rate that are associated with voice and data traffic, as well as the impact each type of traffic has on the other. A variety of approaches are available for channel access in data networks. Depending on the nature of the traffic, either contention-based or contention-free schemes or their hybrids can be used. Many comparative discussions of channel-access protocols are found in the literature (e.g., Refs. 1 and 2) and we do not discuss their merits here. However, the problem of channel access in integrated radio networks has not received much attention.

In this report we introduce and analyze a family of new protocols for integrated voice/data communication that are primarily well suited to satellite networks or to high-rate terrestrial wireless networks. These protocols are modifications of the Interleaved-Frame Flush Out (IFFO) protocols for data traffic, which were introduced by Wieselthier and Ephremides [3, 4] a decade ago. After a brief discussion of the major issues associated with channel access in integrated radio networks, we review the IFFO protocols and the mathematical model used to describe them. These protocols are characterized by a frame length that adapts to bursty channel traffic, resulting in very high efficiency.

We then consider variations of the IFFO protocols (still considering only data traffic) in which the frame length is kept constant. These are known as the Interleaved-Frame Fixed-Length (IFFL) and Non-Interleaved-Frame Fixed-Length (NIFFL) schemes. The property of constant frame length is desirable for voice traffic, which is generally characterized by the need for near-real-time delivery but, more importantly, for constant delay. By using this framework, we then extend the NIFFL protocols for operation in integrated networks by incorporating a movable-boundary mechanism to share the channel between voice and data traffic; the new protocols are called the Voice/Data NIFFL (VD-NIFFL) protocols.

Markov chain models have been constructed for these protocols. An exact analysis of one version of VD-NIFFL is presented that uses a pure reservation scheme for data traffic. We exploit special features of the system to reduce complexity, thereby making numerical solution possible. Analytical models are developed for the evaluation of data-packet delay under both fixed- and movable-boundary versions of VD-NIFFL, as well as for the fraction of voice calls that are blocked. Extensive performance results are presented.

2. BOUNDARY SCHEMES FOR VOICE/DATA MULTIPLEXING

The goal of voice/data integration is to share network resources efficiently between these two classes of traffic while satisfying the performance requirements of both. Voice traffic is characterized by the need for delivery as a continuous stream in near-real time. More importantly, the delay must

Manuscript approved August 20, 1991.

be nearly constant throughout the duration of each talkspurt. It is assumed that buffering of voice is not permitted.* Calls arc blocked if channel resources are not immediately available; acceptance of a voice call requires a continuous commitment of channel resources (e.g., a time slot in every frame or a fixed portion of the bandwidth for frequency division systems) for the entire duration of the call. In contrast, data packets are characterized by a need for very low packet-error probability but do not require real-time delivery. Delay requirements for data depend on the nature of the traffic and may be different for different classes of traffic in the network. Buffering of data packets is permitted.

The need to support the requirements of voice traffic results in the need for contention-free channel access once a call has been set up. Reservation schemes, which can maintain throughput levels near channel capacity, are the logical choice for voice calls. Many reservation schemes are proposed in the literature; most of them are modifications of the demand-assignment scheme originally proposed by Roberts [5].

Movable-Boundary Schemes

Most studies of voice/data integration have addressed this problem from the perspective of multiplexing at a single node. Our discussion focuses on the "movable-boundary" scheme, which has emerged as the prime hybrid system of integrated switching (see e.g., Ref. 6). As we demonstrate in this report, it is straightforward to extend the boundary concept from the realm of multiplexing to that of channel access.

The boundary scheme is based on a time-division multiple-access (TDMA) frame structure (Fig. 1). The fixed-length TDMA frame is partitioned into two compartments; voice is circuit-switched in one and data is packet-switched in the other. The boundary between the two compartments can be either fixed or movable. Under the fixed-boundary scheme, each commodity (voice or data) remains confined to its allocated compartment at all times. Data traffic can be transmitted only in those slots that are a priori allocated to data, and similarly voice traffic is restricted to those slots that are allocated to voice. Under this scheme, we may have the undesirable situation in which slots in the voice compartment remain idle while data packets are required to remain in queue. To alleviate this inefficiency, the movable-boundary scheme has been proposed. In this scheme data traffic is allowed to use any idle slots of the voice compartment, resulting in higher bandwidth utilization; however, voice traffic is not allowed to use unused data slots.

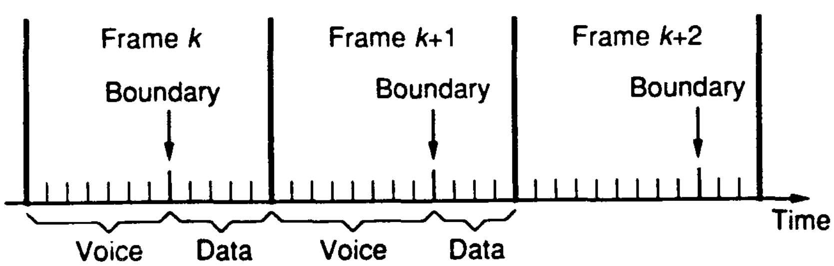


Fig. 1 — Movable-boundary channel

The acceptance of a voice call by the communication system implies a long-term commitment of a channel (in this case a periodically recurring TDMA slot) to support the call for its entire duration. It is generally assumed that a voice call cannot be interrupted once it is assigned a channel. Data traffic requires only a short-term commitment, i.e., one packet at a time.† Since each data packet

*Even if voice buffering is permitted, the requirements for continuity and constant delay do not change.

†Although data messages can consist of more than one packet, each packet of a multipacket message can be treated separately by the network since there is no need for uniformity of delay.

occupies only one slot at a time, data traffic does not interfere with voice traffic, even under the movable-boundary scheme. The voice slot that was borrowed for data reverts to its original status as a voice slot immediately when needed for this purpose. In contrast, if a voice call were permitted to use a slot from the data compartment, this slot would be unavailable to data traffic for the entire duration of the call.

Movable-boundary schemes invite the use of dynamic optimization techniques that adapt to channel traffic. Based on traffic, the position of the boundary can be chosen to optimize system performance. However, in this report we limit our discussion of optimization to the simpler case in which a static or open-loop control is implemented.

Channel Access Considerations

Clearly, similarities exist between the channel-access problem and the multiplexing problem. However, there are also some important differences. In the multiplexing problem the goal is the optimal sharing of a contention-free channel that has already been established between the source and destination. A single node makes the decisions of which voice and data streams to multiplex into a single waveform. The link throughput can be maintained at its maximum value as long as the node has something to transmit over it. For example, some studies of multiplexing schemes have taken advantage of the silent periods that are inherently a part of voice traffic by permitting data traffic to use such gaps (see e.g., Ref. 7).

In the channel-access problem, the goal is the optimal use of a channel by a distributed population of users who can interfere with each other's transmissions. The channel-access problem is considerably more complicated than the multiplexing problem because not all transmissions are successful and because distributed operation is often necessary to obtain robust and efficient performance. Although it would be straightforward for any particular node to use the silent periods in its own voice stream for data transmission, it would be difficult to share these time slots with another node because of the randomness in the voice process and because of the delay involved in sensing speech gaps at another node, particularly when the propagation delay is large. Thus we do not consider the possible exploitation of silent periods in this report.

3. EARLIER STUDIES OF CHANNEL ACCESS IN INTEGRATED NETWORKS

The problem of channel access in integrated radio networks has not received much attention in the literature. Recently, Suda et al. [8] and Wu and Li [9] studied protocols for access to satellite channels. These schemes are characterized by a fixed-length frame structure that contains reservation channels and information channels (some of which are allocated for voice and the remainder for data). Voice is handled on a reservation basis in both of these studies. Once a reservation for a voice call is made successfully, one slot per frame is allocated to the call until its completion. There are a number of differences in the two models, however, particularly in the channel-access mechanism for data. Nevertheless, these schemes bear a strong similarity to the ideas developed in Ref. 3 for the IFFO protocols, which inspired the introduction of the new protocols that form the main contribution of this report.

It is also worthwhile to mention Soroushnejad and Geraniotis's studies of a movable-boundary scheme that exploits code-division multiple-access (CDMA) techniques [10 and 11]. However, we do not consider CDMA approaches in this report.

4. IFFO PROTOCOLS FOR DATA-ONLY OPERATION

In this section we describe the communication system for data-only operation. We present first-order Markov chain models to characterize the IFFO protocols, and we describe the procedure used to obtain equilibrium performance results. It is then straightforward to extend the model to integrated networks, which are discussed in Section 7. We consider M ground-based users (terminals) that communicate among themselves via a transponder that broadcasts all messages it receives to all members of the user population.

Data traffic consists of fixed-length packets, each of which requires one time "slot" for transmission. We define R to be the round-trip propagation delay, measured in terms of slot durations. Let us first consider the case of a geosynchronous (stationary) satellite, for which the round-trip propagation delay is approximately 0.27 second; e.g., for a data rate of 50,000 bits/s and a packet length of 1125 bits, we get a value of $R = 12$ (this is the value that was used in Ref. 3). Because currently available data rates can be up to three orders of magnitude greater, the study of considerably larger values of R and/or packet length is also now appropriate. For example, for a data rate of 5 Mbits/s and the same packet length, we now have $R = 1200$ for geosynchronous satellites. In fact, we may have $R >> 1$ even in terrestrial systems. For example, a communication range of 100 km results in a one-way propagation delay of 0.333 ms, and hence a round-trip propagation delay of 0.667 ms. For a data rate of 50 Mbits/s and a packet length of 5000 bits (hence 10,000 packets per second), we get $R = 15$. (However, it will be necessary to incorporate guard times to maintain constant delay throughout the network by compensating for different propagation path lengths.) In this report, we concentrate primarily on the case of $R > 1$, and we present performance results for three values of R, namely 6, 12, and 120. In Section 12 we address the case of $R < 1$, which is more representative of many terrestrial military applications. Such systems are generally characterized by low data rates (e.g., 2400 b/s), and thus typically require packet durations of tens to hundreds of ms.

It is assumed that each user has an infinite buffer in which to store the arriving packets, which are assumed to form a Bernoulli process with rate λ in every slot. The total arrival rate is, therefore, $M\lambda$ packets per slot, which is equal to the throughput rate under stable operation since no packets are rejected.

Markov Chain Model for the IFFO Protocols

The basic structure for the IFFO protocols (Fig. 2) is a reservation structure in which the unreserved slots can be used for transmission on a contention basis. The first slot of each frame, which consists of M "minislots" that are exclusively allocated to the M terminals in (contention-free) TDMA fashion, is known as the status slot. It is used by each of the terminals to reserve a transmission slot for each of the packets that were generated in the previous frame.*

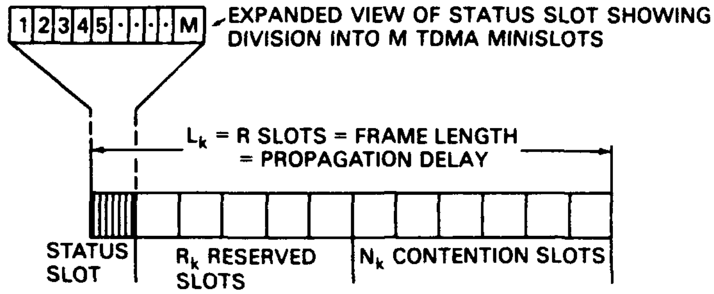
It is assumed that all reservation minipackets are received successfully by all terminals following the round-trip propagation delay of R slots. We define:

R_k = total number of reserved slots in frame k;

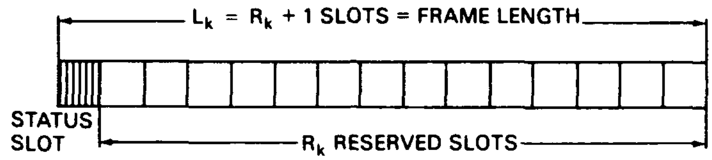
N_k = total number of contention (unreserved) slots in frame k;

L_k = total length (in slots) of frame k.

*Clearly, the number of terminals that can be accommodated by this protocol is limited by the number of minislots that can be established in one slot duration. These minipackets could be quite short because the only information they would have to deliver is the number of packets that arrived at the terminal during the previous frame. Alternatively, we may view the system as one with a large number of users but a relatively small number of concentrators, each of which forwards the reservations and packets of several users.



CASE 1: WITH CONTENTION SLOTS, i.e., $R_k \leq R-2$



CASE 2: WITH NO CONTENTION SLOTS, i.e., $R_k \geq R-1$

Fig. 2 — Frame structure for IFFO protocols

Thus the status slot is followed by R_k reserved slots. It is required that each frame have a length of at least R slots to ensure that the reservation information generated at the beginning of the frame is received before the end of the first slot of the next frame. If $R_k < R - 1$, the remaining $N_k = R - 1 - R_k$ slots in the frame are used for transmission on a contention basis, as in the slotted-ALOHA protocol [1, 2, and 12]. If $R_k > R - 1$, additional slots are added to accommodate all of the reservations. Therefore,

$$N_k = \max(R - 1 - R_k, 0), \tag{1}$$

and

$$L_k = \max(R_k + 1, R) = 1 + R_k + N_k. \tag{2}$$

Since the frame length expands to accommodate all packets for which reservations have been received, and since there is one slot of overhead per frame regardless of frame length, throughput rates arbitrarily close to one packet per slot can be realized. However, packet delay increases rapidly as throughput approaches one.

The quantity of interest that needs to be tracked is R_k, which evolves as a first-order Markov chain. Reference 3 provides a complete discussion of the dynamics. Here we summarize the derivation and show the resulting transition probabilities.

Figure 3 illustrates the operation of the IFFO protocols. Each packet arriving in frame k is known as a k-packet. Reservation minipackets for all k-packets are transmitted in the first slot of frame $k + 1$. Although reservations are transmitted for all k-packets, some of these reservations may not be needed because of the possibility of successful transmission in contention slots. It is easy to cancel the unneeded reservations because knowledge of the outcome of all contention transmissions in frame k will be available to all users no later than the beginning of frame $k + 2$.

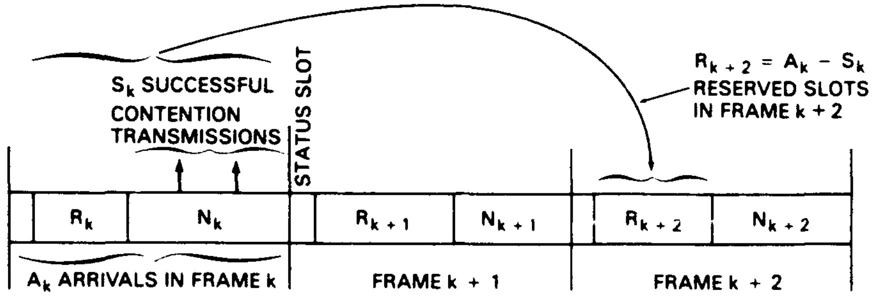


Fig. 3 — Operation of the IFFO protocols, showing transmission procedure for frame-k arrivals

By observing the "downlink" channel traffic, the terminals can count the number of packets successfully transmitted by each terminal in the contention slots of frame k. Knowledge of this information (which is possessed by all terminals because of the broadcast nature of the channel, and which arrives before the corresponding reservation minipackets) can then be used to cancel unneeded reservations. No central controller is needed. We only need an agreed-upon protocol for scheduling transmissions, given that each user knows the communication needs of every other user. As long as every terminal knows the allocation rule being followed, the protocol will be uniquely defined.* Let

A_k = total number of packet arrivals in frame k, summed over all terminals;

S_k = number of successful contention transmissions in frame k.

Then,

$$R_{k+2} = A_k - S_k. \tag{3}$$

Note that R_{k+2} is totally independent of A_{k+1}, L_{k+1}, and R_{k+1}. It depends only on R_k and on what happens during frame k. Thus we can split the process $\{R_k\}$ into two interleaved Markov chains, which are denoted $\{R_{2j}\}$ and $\{R_{2j+1}\}$. The reserved slot process in even-numbered frames is independent of the reserved slot process in odd-numbered frames. These processes have identical statistics and can be analyzed separately.

IFFO Protocol Versions

There are several versions of the IFFO protocols, each of which is characterized by a different transmission procedure in the unreserved slots:

1. *Pure Reservation IFFO (PR-IFFO)*: The unreserved slots are not used for contention; they simply remain idle and wasted. All packets that arrive in frame k are transmitted in the reserved slots of frame $k + 2$.

2. *Fixed Contention IFFO (F-IFFO)*: The transmission policy depends on the slot number in which packets arrive.

   - A packet arriving in slot n, for $n \in (R_k + 1, R - 1)$, will be transmitted in slot $n + 1$, i.e., in each contention slot, each terminal will transmit the packet that may have

*Alternatively, an intelligent satellite or central controller could process the reservations and recognize successful contention transmissions, and then broadcast a transmission schedule to the users during the status slot instead of simply repeating the reservations. Doing so would eliminate the possibility of collisions that may result from inconsistencies in the database of reservations, which may be caused by errors on the downlink.

arrived in the previous slot. Each colliding packet will be retransmitted in a reserved slot in frame $k + 2$.

- All packets arriving during the first R_k slots of frame k will not be allowed to contend because of the high risk of collision, and will be assigned reserved slots for transmission during frame $k + 2$.

- All packets arriving during slot R (i.e., the last slot in the frame) will be assigned reserved slots in frame $k + 2$.

3. *Controlled Contention IFFO (C-IFFO)*: In this version, the transmission procedure is state-dependent. In each contention slot of frame k, a nonempty terminal transmits a packet with a probability that is a function of the number of packets present at that terminal, the slot number, and R_k. This transmission probability, which depends only on local information and thus permits totally distributed operation, is chosen to maximize throughput. References 3 and 4 provide a complete description and approximate analysis of this protocol, including a discussion of optimization issues.

It is apparent from their definitions that F-IFFO and C-IFFO provide better performance than PR-IFFO, since the hybrid schemes use slots that would have otherwise been left idle for transmission on a contention basis. Performance results [3, 4] demonstrate that the improvement is substantial at low to moderate throughput rates. However, at high throughput rates few packets are able to take advantage of the contention mode of operation, and PR-IFFO provides a close upper bound on expected delay that can be achievable under the hybrid schemes. This upper bound becomes increasingly tight as throughput increases. It is also significant to note that the performance of F-IFFO, which uses a very simple transmission policy in contention slots, performs almost as well as a version of C-IFFO, under which the transmission probabilities have been optimized. References 3 and 4 also show and that F-IFFO performs better than several other hybrid protocols over a wide range of throughput rates.

Although improved performance can be expected under the hybrid versions of the new schemes for integrated voice/data operation as well, we limit our discussion in the body of this report primarily to the pure-reservation versions of the new schemes. This permits us to emphasize those features of the analysis that relate to integrated voice/data operation, without introducing the additional complexity associated with the hybrid schemes. In Appendix A we discuss the fixed-contention versions of IFFO and the new schemes.

Transition Probability Matrix for PR-IFFO

For PR-IFFO, since $R_{k+2} = A_k$,* it is easy to see that the elements of the transition probability matrix for R_k can be written as†

$$p_{ij} \triangleq Pr(R_{k+2} = j \mid R_k = i) = \begin{cases} \binom{MR}{j} \lambda^j (1 - \lambda)^{MR - j}, & 0 \le i \le R - 1 \\ \binom{M(i+1)}{j} \lambda^j (1 - \lambda)^{M(i+1) - j}, & i \ge R - 1. \end{cases} \quad (4)$$

Here we assume symmetric traffic, i.e., each user generates a data packet with probability λ in every slot. A similar expression for F-IFFO is provided in Appendix A.

*Recall that the data users are assumed to have infinite buffers, thus all arrivals are accepted into the system.
†Computational issues associated with the evaluation of these quantities are discussed in Appendix B.

Obtaining Equilibrium Results for IFFO Protocols

The equilibrium probability mass function (pmf) for the number of reserved slots per frame R_k is needed to evaluate the expected time spent in the system per packet. References 4 and 13 provide expressions for expected packet delay.

An equilibrium pmf for R_k does, in fact, exist under the IFFO protocols, as long as the input rate is less than one packet per slot. In this case, R_k, which is irreducible and aperiodic, is easily shown to be ergodic by Foster's theorem [14].

Note that the chain that describes R_k has an unbounded (hence, infinite) number of states. To obtain a numerical solution, we truncate the probability vector and transition probability matrix to some finite dimension N. The value of N can be chosen sufficiently large that the effect of truncation error is small.

Method Used to Determine the Equilibrium pmf π: The equilibrium pmf π (a row vector) must satisfy the matrix equation:

$$\pi = \pi \mathbf{P}, \tag{5}$$

where $\mathbf{P}$ is the transition probability matrix with elements $p_{ij} \triangleq Pr(R_{k+2} = j \mid R_k = i)$. Instead of solving the N equations in N unknowns, it is computationally preferable to use the iterative procedure of relaxation, i.e.,

$$\pi = \lim_{n \to \infty} \pi(0)\mathbf{P}^n \tag{6}$$

where $\pi(0)$ is an arbitrary initial pmf, and the iteration is stopped when there is sufficiently small change in each of the elements of π. Computational issues associated with this operation are discussed in Appendix B.

5. INTERLEAVED-FRAME FIXED-LENGTH (IFFL) SCHEMES

The IFFO protocols are characterized by a frame length that adapts to channel traffic. This adaptive feature guarantees that all k-packets are successfully transmitted not later than the end of frame $k + 2$; this is the flush-out feature of these schemes that, indeed, motivated them and that results in very high throughput and excellent delay performance. In this section we consider a variation of these protocols under which the frame length is kept fixed at $L_k = L$ slots, where $L \geq R$. In certain applications (e.g., voice/data integration) it is desirable to keep a constant frame length, although doing so reduces the achievable throughput of the protocol.

We call these schemes the Interleaved-Frame Fixed-Length (IFFL) schemes. Operation is the same as that of IFFO, except that when R_k is greater than L, packets that cannot be accommodated in the current frame are delayed until frame $k + 2$, as shown in Fig. 4, at which point they are again subject to further delays if there is again a large backlog. The number of such "excess packets" in frame k is

$$\tilde{R}_k = \max\{(R_k - L + 1), 0\}. \tag{7}$$

Clearly, these protocols violate the flush-out condition, and therefore do not belong to the class of IFFO protocols. We consider PR-IFFL and F-IFFL versions of these schemes, whose definitions follow from those given earlier for the IFFO schemes. System evolution is again characterized by the first-order Markov chain R_k. However, note that a slight reinterpretation of R_k is needed. It was

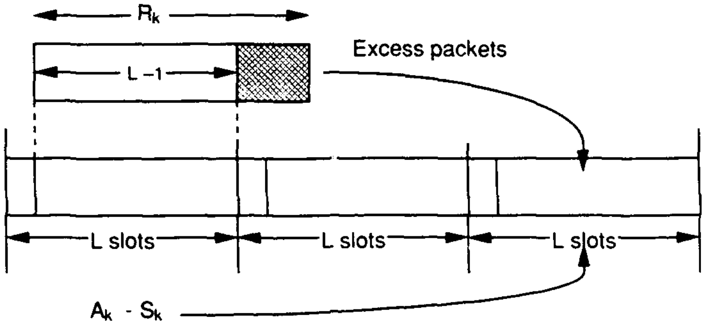


Fig. 4 — Operation of IFFL

previously defined as the number of reserved slots in frame k. Because the excess packets are delayed until frame $k + 2$, R_k must now be interpreted as the number of packets for which reservations are needed at the beginning of frame k (including the excess packets, which have to be delayed). We have

$$R_{k+2} = A_k - S_k + \tilde{R}_k, \tag{8}$$

where $S_k = 0$ for PR-IFFL. By straightforward modification of the expressions for PR-IFFO, the elements of the transition probability matrix for PR-IFFL can now be written as

$$p_{ij} \triangleq Pr(R_{k+2} = j \mid R_k = i) = \begin{cases} \binom{ML}{j} \lambda^j (1-\lambda)^{ML-j}, & 0 \le i \le L-1 \\ \binom{ML}{j-i+L-1} \lambda^{j-i+L-1} (1-\lambda)^{ML-(j-i+L-1)}, & i \ge L-1, \end{cases} \tag{9}$$

where $\binom{m}{n} = 0$ for $n < 0$. Appendix A provides similar expressions for F-IFFL.

6. NON-INTERLEAVED-FRAME FIXED-LENGTH (NIFFL) SCHEMES

Under the IFFO and IFFL schemes, the system state in even-numbered frames is independent of that in odd-numbered frames. Thus it is possible for the even-numbered slots to build up large backlogs (high values of R_k) while the odd-numbered slots are lightly loaded. Since the IFFO schemes flush out all k-packets by the end of frame $k + 2$, no inefficiency arises from this behavior. However, under IFFL, when $R_k > L - 1$, the $\tilde{R}_k$ excess packets will be postponed to frame $k + 2$. If any unreserved slots are present in frame $k+1$, it would be advantageous to transmit some or all of these excess packets in those slots; there is no need to postpone them until frame $k+2$ since reservations for them have already been received.

To address this situation we consider a variation of the IFFL protocols that we call the Non-Interleaved-Frame Fixed-Length (NIFFL) protocols. We show that the PR-NIFFL version is again characterized by an underlying first-order Markov chain, and thus can be evaluated by using techniques similar to those used for the IFFO and IFFL schemes. However, the description of F-NIFFL (the hybrid version, which uses the unreserved slots for contention transmission) requires a second-order Markov chain. This makes performance evaluation considerably more difficult, as is discussed in Appendix A. Figure 5 illustrates system evolution for the NIFFL protocols. We note that R_{k+2} can be decomposed as follows:

$$R_{k+2} = R_{k+2}^{(k)} + R_{k+2}^{(k+1)}. \tag{10}$$

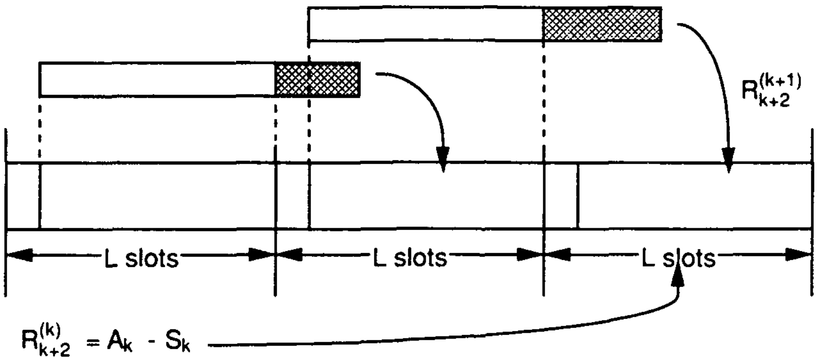


Fig. 5 — Operation of NIFFL

The quantity $R^{(k)}_{k+2}$ is the number of k-packets that are included in R_{k+2}, i.e., all arrivals in frame k, except those that were transmitted successfully in the contention slots of frame k. Thus we have

$$R^{(k)}_{k+2} = A_k - S_k. \tag{11}$$

The quantity $R^{(k+1)}_{k+2}$ is the contribution to R_{k+2} from the R_{k+1} packets in queue at the beginning of frame $k+1$, i.e., it is the number of excess packets that are carried over from frame $k+1$. Thus we have

$$R^{(k+1)}_{k+2} = \tilde{R}_{k+1} = \max\left\{[R_{k+1} - (L-1)],\ 0\right\}. \tag{12}$$

The Markov chain now is the pair (R_{k+1}, R_k), and thus we need

$$p_{ij-jm} \triangleq Pr(R_{k+2} = m,\ R_{k+1} = j \mid R_{k+1} = j,\ R_k = i). \tag{13}$$

A brute-force system description would require a state probability vector $Pr(R_{k+1}, R_k)$ of dimension N^2, where N is the truncation value discussed earlier. We can write

$$Pr(R_{k+1}, R_k) \triangleq \left(\mathbf{q}_0 \quad \mathbf{q}_1 \quad \cdots \quad \mathbf{q}_{N-1}\right), \tag{14}$$

where $\mathbf{q}_j$ is the row vector whose entries are

$$q_{ji} \triangleq Pr(R_{k+1} = j,\ R_k = i). \tag{15}$$

Thus a transition probability matrix of size $N^2 \times N^2$ would be needed. The dimensions of the problem can be reduced somewhat by making the observation that R_{k+1} includes all of the excess packets contained in R_k. Thus, given R_{k+1}, the exact value of R_k is needed only if it is less than $L-1$. We define an aggregate state $R_k = L-1$ that actually contains all states for which $R_k \geq L-1$. Now the state (R_{k+1}, R_k) can be described by a vector of dimension NL rather than N^2. The state (R_{k+2}, R_{k+1}) still requires a vector of dimension N^2, however. The resulting transition probability matrix is of dimension $NL \times N^2$. This is still of undesirably large size.

Appendix A describes how the structural properties of F-NIFFL can be exploited to decompose the problem into a number of smaller problems, without requiring any approximations. In particular, our approach requires N matrices of dimension $L \times N$. Although the resulting problem is still rather complicated and computationally intensive, it is considerably smaller than before. Here we demonstrate how the properties of the PR-NIFFL protocol can be exploited to permit its exact evaluation by using a first-order Markov chain, thereby simplifying the model greatly without requiring the use of approximations.

Under PR-NIFFL, $R_{k+2}^{(k)} = A_k$ because $S_k = 0$ (since there are no contention transmissions in a pure reservation system). Thus

$$R_{k+2} = A_k + \tilde{R}_{k+1}. \tag{16}$$

Since the frame length is constant, A_k does not depend on R_k. It is binomially distributed with parameter λ over ML trials. Thus the system description for PR-NIFFL can be reduced to a first-order Markov chain as follows:

$$p_{ij} \triangleq Pr(R_{k+2} = j \mid R_{k+1} = i) = \begin{cases} \binom{ML}{j} \lambda^j (1-\lambda)^{ML-j}, & 0 \le i \le L-1 \\ \binom{ML}{j-i+L-1} \lambda^{j-i+L-1} (1-\lambda)^{ML-(j-i+L-1)}, & i \ge L-1. \end{cases} \tag{17}$$

Note that this expression is identical to that for PR-IFFL, except that R_k is replaced here by R_{k+1}.

7. NIFFL PROTOCOLS FOR INTEGRATED VOICE/DATA SYSTEMS

The communication systems that gave rise to the models discussed thus far can be modified slightly to handle voice traffic in addition to data packets. This is an important extension in communication system design and represents our main objective here. A customary model for voice assumes that voice calls are generated by idle users according to a Bernoulli process, and that they are geometrically distributed in length; thus the probability that a call is completed in any particular frame is also a Bernoulli process. The time constants associated with voice traffic are considerably larger than those associated with data traffic; voice calls typically last from tens to hundreds of frames. This difference in time durations has played a key role in the development of approximate system models for this protocol [15, 16, and 17]. In this report we consider only the exact model, since extensive performance results based on it are now available. The voice-call process is discussed in detail in Section 8, where we discuss the conditions that lead to a first-order Markov chain description of the system.

To accommodate the needs of both voice and data traffic, we consider channel-access protocols under which a reservation scheme is used for voice traffic and NIFFL is used for data. We call these the Voice/Data NIFFL (VD-NIFFL) protocols.

Under these schemes, once a voice call is accepted by the system, it is guaranteed access to one slot in each frame until its completion.* A fixed-length frame structure is necessary to accommodate the real-time requirements of voice traffic. It is appropriate for both satellite and ground-radio environments, and for any values of R greater than or less than 1. The standard idea of a "boundary" mechanism is used to partition each frame between voice and data operation (Fig. 6).

*Here we implicitly assume that the slot length has been selected in conjunction with the frame length (which is greater than or equal to the propagation delay) so that the voice burst rate results in the required symbol rate needed for real-time voice transmission. For example, in applications where $R << 1$, any frame length will guarantee that reservations are received prior to the beginning of the next frame.

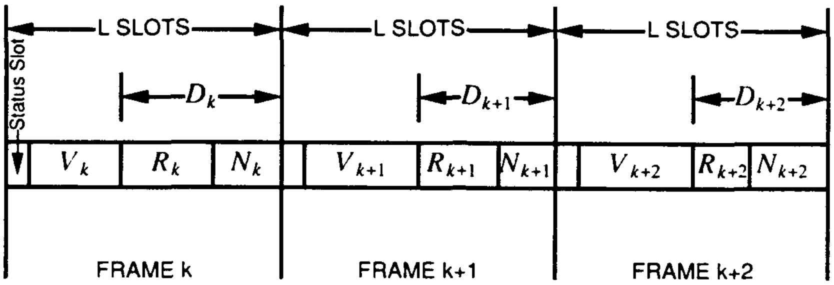


Fig. 6 — Frame structure for VD-NIFFL protocols

Although voice and data users both make their reservations on a contention-free basis, voice and data traffic are handled in entirely different ways by the network. Voice calls are accepted by the system as long as the total number of calls in progress simultaneously does not exceed some specified value V_{max}, which must be less than L. If a slot is not available for a new call, the call is assumed to be lost; voice calls are not buffered. Under the movable-boundary implementation, the slots not used by voice calls (including empty slots in the voice portion of the frame) are used for data traffic, which is transmitted by using one of the NIFFL protocols. Since the decision to accept new voice calls depends only on whether or not the threshold V_{max} is exceeded, voice traffic is unaffected by data traffic; however, the operation of the data protocol is dependent on voice traffic because data traffic is permitted to use unneeded slots in the voice portion. Thus this problem is similar to variable-service-rate queueing systems in which the service rate depends on another process.

Under a fixed-boundary implementation of the VD-NIFFL protocols, a fixed number of slots is available for data traffic in every frame. In this case the data-packet process is independent of the voice-call process, and the overall integrated system functions as two totally independent subsystems, one for voice and the other for data. Our primary interest is in the movable-boundary implementation because of its ability to provide improved performance and because its analysis is considerably more challenging. However, we do indicate how the model can be simplified for the fixed-boundary case.

Our model can be extended to consider systems in which the decision to accept new voice calls also depends on the system backlog (i.e., the value of R_k). However, the analysis of such systems is considerably more difficult and is not addressed here. Multiplexing (although not channel access) systems incorporating this feature were studied by Viniotis and Ephremides [18, 19].

As shown in Fig. 6, the first slot of every frame is once again the status slot, during which each terminal transmits its reservations for packets that arrived in the previous frame. Under a movable-boundary implementation, the next V_k slots are reserved for voice traffic, where V_k is the number of voice calls in progress at the beginning of slot k. The remainder of the frame consists of D_k data slots, where

$$D_k = L - 1 - V_k. \tag{18}$$

(Under a fixed-boundary implementation, $D_k = L - 1 - V_{max}$, independent of V_k.) As with the protocols designed purely for data, R_k is the number of data packets for which reservations are needed at the beginning of frame k. Whenever $R_k < D_k$, N_k^v slots are available foı contention transmission, where

$$N_k^v = \max(D_k - R_k, 0) = \max(L - 1 - v - R_k, 0) \tag{19}$$

for each particular value of $V_k = v$. Whenever $R_k > D_k$, the excess packets are delayed until frame $k+2$. Operation of the data portion of VD-NIFFL can thus be viewed as that of NIFFL with a variable number of slots (D_k) available for data traffic, where D_k depends on V_k. In contrast, under data-only operation of NIFFL exactly $L - 1$ slots are available for data in each frame.

8. AN EXACT MARKOV CHAIN MODEL FOR VD-NIFFL

The VD-NIFFL protocol can be characterized by the Markov chain (R_k, V_k), which has transition probabilities $Pr(R_{k+1}, V_{k+1} \mid R_k, V_k)$. The development of a Markov chain model for the VD-NIFFL protocols has taken into account the dependence of data traffic on voice, whereas voice is independent of data. A brute-force approach would consider a probability vector containing all possible pairs of V_k and R_k. The maximum value of V_k would be the threshold value V_{max}; R_k would have a maximum value of $N - 1$ as in the evaluation of the data-only NIFFL protocols. Thus a transition probability matrix of dimension $(V_{max} + 1)N \times (V_{max} + 1)N$ would be needed (e.g., for $V_{max} = 6$ and $N = 700$, these matrices would be 4900×4900). However, not all transitions are possible, and dramatic reductions in the number of computations needed can be made by decomposing the problem into separate voice and data portions. We do this by recognizing that the voice-call process does not depend on the data-message process.* Thus

$$Pr(R_{k+1}, V_{k+1} \mid R_k, V_k) = Pr(R_{k+1} \mid R_k, V_k)\, Pr(V_{k+1} \mid V_k). \tag{20}$$

The transition from R_k to R_{k+1} depends on V_k in a movable-boundary system (because V_k determines D_k), but not on V_{k+1}.† The transition from V_k to V_{k+1} does not depend on R_k or R_{k+1}. We first note that these observations simplify the evaluation of the transition probability matrix. Actually, a much greater benefit is realized. It is demonstrated below that the transitions corresponding to the data process can be considered separately for each value of V_k. Thus it is not necessary to perform the iteration with the huge transition probability matrix that characterizes the evolution of the complete voice/data state description. The evaluation of system performance can be decomposed into a number of smaller problems that are of manageable size.

We observe that the transitions from frame k to frame $k + 1$ can be modeled as a two-step process. Data transitions are considered first. Given $Pr(R_k, V_k)$, we first determine $Pr(R_{k+1}, V_k)$. This requires a different transition probability matrix for each value of V_k, as explained below. This operation can be expressed as follows:

$$Pr(R_{k+1} = j,\ V_k = v) = \sum_{i=0}^{N} Pr(R_{k+1} = j \mid R_k = i, V_k = v)\, Pr(R_k = i, V_k = v), \quad 0 \le j \le N - 1. \tag{21}$$

Next, the voice transitions are considered. Given $Pr(R_{k+1}, V_k)$, we determine $Pr(R_{k+1}, V_{k+1})$. Since the voice transitions are independent of the data traffic, the same transition probability matrix is used for all values of R_{k+1}. Thus, the following is evaluated:

$$Pr(R_{k+1} = j, V_{k+1} = w) = \sum_{v=0}^{V_{max}} Pr(V_{k+1} = w \mid V_k = v)\, Pr(R_{k+1} = j, V_k = v). \tag{22}$$

The equilibrium distribution of the system state is determined by repeating this two-step iteration until convergence is achieved. We emphasize that this model is exact. The characterization of the system by smaller transition probability matrices has been achieved by exploiting specific structural properties of the Markov chain, not by making simplifying approximations (other than the truncation of the transition probability matrices associated with data transitions to a finite size). We now discuss the details of our model.

*This is not true for systems in which the decision on whether or not to accept a voice call is permitted to depend on R_k, however.

†In a fixed-boundary system, data transitions are independent of the voice process. Thus $Pr(R_{k+1} \mid R_k, V_k) = Pr(R_{k+1} \mid R_k)$, and hence $Pr(R_{k+1}, V_{k+1} \mid R_k, V_k) = Pr(R_{k+1} \mid R_k)\, Pr(V_{k+1} \mid V_k)$.

Data Transitions

We first consider the data transitions. Corresponding to each value of V_k is an $N \times N$ transition probability matrix for the data message process with elements

$$p_{ij}^{v} \triangleq Pr(R_{k+1} = j \mid R_k = i, V_k = v). \tag{23}$$

These transition probabilities are easily obtained from those for the NIFFL protocols. Under NIFFL, $L-1$ slots are available for packet transmission in each frame. Under VD-NIFFL, this number is reduced to $D_k = L-1-V_k$. Thus for each value of $V_k = v$ we replace $L-1$ by $L-1-v$. For a movable-boundary implementation of PR-VD-NIFFL this yields

$$p_{ij}^{v} = \begin{cases} \binom{ML}{j} \lambda^{j}(1-\lambda)^{ML-j}, & 0 \le i \le L-1-v \\ \binom{ML}{j-i+L-1-v} \lambda^{j-i+L-1-v}(1-\lambda)^{ML-(j-i+L-1-v)}, & i \ge L-1-v. \end{cases} \tag{24}$$

Thus, at each iteration, $(V_{max} + 1)$ matrix multiplications (each of size $N \times N$, one for each value of v) must be carried out to determine the data transitions.

Under fixed-boundary implementations of these protocols, p_{ij}^{v} is independent of v, hence $p_{ij} = p_{ij}^{v} = p_{ij}^{V_{max}}$ (i.e., replace v by V_{max} in the above equations) for all values of v. Thus the two-step iteration procedure is not needed. The equilibrium distribution of the data-packet process can be determined totally independently of the voice-call process by following the iteration procedure discussed earlier for the data-only protocols.

Voice Transitions

Next, we consider the voice transitions. The probability that a new call is generated at a terminal during any particular frame is denoted as λ_V. We can view the M_V voice terminals as concentrators, each of which can support several (up to V_{max}) voice calls simultaneously. Each terminal is able to generate one new voice call in any given frame, with probability λ_V, independent of the number of voice calls it is already supporting. The probability that an ongoing call completes service during any particular frame is denoted μ_V.

The assumption that the probability of call generation is independent of the number of calls in progress at that terminal is reasonable for large user populations. The probability of generating more than one call at a terminal in a frame is sufficiently small for reasonable system parameters that it can be neglected.

Reservations are made during the status slot, in the same manner as those for data traffic. Thus, if a voice call arrives during frame k, its reservation will be transmitted during the first slot of frame $k + 1$, and, if available, a slot will be reserved for it beginning in frame $k + 2$. A voice call will be accepted if and only if the total number of voice calls in the system will not exceed the threshold V_{max}. When a call is blocked, it is dropped from the system.

An end-of-message (EOM) indicator is transmitted during the last frame of a voice call to indicate that the slot is no longer needed in subsequent frames. The way in which this EOM is implemented has a critical impact on the implementation and analysis of the VD-NIFFL protocols. The normal way to implement an EOM is to simply insert it at the end of the last packet in the call. No additional packet (or frame) would be needed because the EOM, which would consist of just a few bits, could be incorporated into the tail information transmitted at the end of the last packet. However, since the frame length is assumed to be equal to the propagation delay, an EOM transmitted in frame k is not received until the end of the corresponding slot of frame $k + 1$; thus this slot would remain idle during frame $k + 1$, and would not become available to a new voice call until frame $k + 2$. In addition to the obvious inefficiency associated with letting the slot remain idle in frame $k + 1$, the mathematical description of the voice-call process is complicated by the fact that a second-order Markov chain would be needed to describe it, since the effects of a call completion could not be exploited until two frames later.

It might be possible to avoid the wasted slot in frame $k + 1$ by transmitting the EOM in the status slot of frame k, since knowledge that the call is terminating may be available at that time (e.g., if the voice packet transmitted in frame k is formed prior to the end of frame $k - 1$). Alternatively, if the frame length L is at least $R + 1$ slots, knowledge of frame-k departures would be available in time to use the slot in frame $k + 1$. Either of these interpretations permits the development of the first-order Markov chain model for the voice process that we now discuss.*

It is important to note that there is no frame interleaving of the voice call process, since a call occupies a time slot in every frame from its start to its completion. The number of voice calls in frame $k+1$ is

$$V_{k+1} = \min \left\{ (V_k + A_{k-1}^V - U_k), V_{max} \right\}, \tag{25}$$

where A_{k-1}^V is the number of voice-call arrivals in frame $k-1$ and U_k is the number of voice calls that are completed in frame k. It would appear that a second-order Markov chain is needed to describe these transitions, since the above expression involves the arrivals in frame $k-1$, and the number of voice calls and voice completions in frame k. However, A_{k-1}^V is independent of the system state in frame $k-1$ because each terminal is able to generate a new voice call with probability λ_V, independently of the number of voice calls it is already supporting. Consequently, transition probabilities from V_k to V_{k+1} are conditionally independent of V_{k-1} when knowledge of V_k is available. Therefore, the voice-call process can be described by a first-order Markov chain, whose transition probabilities are determined as follows.†

The voice-call process is characterized by a transition probability matrix $\mathbf{S}$ of size $(V_{max} + 1) \times (V_{max} + 1)$. Note that no truncation is required because the number of voice calls cannot exceed V_{max}. To evaluate the elements of $\mathbf{S}$, we first consider the arrival process in frame k. Each of the M_V users (whether or not currently supporting a voice call) generates a new call with probability λ_V. Thus

---

*In Refs. 16 and 17 we presented the analysis of an integrated system that operates under a slightly different set of assumptions. First, it is assumed there that EOMs cannot be transmitted in the status slot and that the frame length equals the propagation delay. As a result of these assumptions, knowledge of a call completion in frame k is not available in time to use the corresponding slot of frame $k + 1$, as discussed above. Second, it is assumed there that each voice terminal can support at most one voice call at a time. Under either of these assumptions, a second-order Markov chain, described by the transition probabilities $Pr(V_{k+2}, V_{k+1} \mid V_{k+1}, V_k)$, is needed to characterize the voice process under these protocols. Although it is not difficult to obtain these transition probabilities, their incorporation into the two-stage voice/data iteration increases the number of computations at each iteration by a factor of $V_{max} + 1$ since a vector of size $(V_{max} + 1)^2$ is now required to describe the voice process instead of a vector of size $V_{max} + 1$.

†A slightly different first-order Markov chain model was developed in Refs. 16 and 17. There, it was assumed that each terminal could support at most one voice call at any given time, an assumption that leads to a second-order Markov chain, as already observed. However, a quasi-static assumption was made, exploiting the fact that the voice-call process changes slowly compared to the data-packet process. In particular, it was assumed there that the probability distribution of A_k^V is the same as that of A_{k-1}^V, resulting in a first-order Markov chain model similar to that which is discussed here.

$$Pr(A_{k-1}^{V} = j \mid V_k = i) = Pr(A_k^V = j \mid V_k = i) = \binom{M_V}{j} \lambda_V^j (1 - \lambda_V)^{M_V - j}, \tag{26}$$

independent of the value of V_k. Now consider the V_k active voice users (not including the new additions), each of which will complete its call with probability μ_V. We have

$$Pr(U_k = j \mid V_k = i) = \binom{i}{j} \mu_V^j (1 - \mu_V)^{i-j}. \tag{27}$$

The elements of the transition probability matrix for the voice-call process are determined as follows. We first consider the case of $V_{k+1} = V_k = v < V_{max}$, which occurs when the number of arrivals in frame $k-1$ is equal to the number of voice call completions in frame k.

$$\begin{aligned} Pr(V_{k+1} = V_k \mid V_k = v) &= Pr(A_{k-1}^V = U_k \mid V_k = v) \\ &= \sum_{u=0}^{v} Pr(A_{k-1}^V = u \mid V_k)\, Pr(U_k = u \mid V_k = v) \\ &= \sum_{u=0}^{v} \binom{M_V}{u} \lambda_V^u (1-\lambda_V)^{M_V - u} \binom{v}{u} \mu_V^u (1-\mu_V)^{v-u}. \end{aligned} \tag{28}$$

Similarly, for $0 < j < V_{max} - v$,

$$\begin{aligned} Pr(V_{k+1} = V_k + j \mid V_k = v) &= Pr(A_{k-1}^V = U_k + j \mid V_k = v) \\ &= \sum_{u=0}^{v} \binom{M_V}{u+j} \binom{v}{u} \lambda_V^{u+j} (1-\lambda_V)^{M_V-u-j} \mu_V^u (1-\mu_V)^{v-u}. \end{aligned} \tag{29}$$

Transitions to $V_{k+1} = V_{max}$ occur whenever $j \geq V_{max} - v$. Thus we have

$$\begin{aligned} Pr(V_{k+1} = V_{max} \mid V_k = v) &= \sum_{j=V_{max}-v}^{M_V} Pr(A_{k-1}^V = U_k + j \mid V_k = v) \\ &= \sum_{j=V_{max}-v}^{M_V} \sum_{u=0}^{v} \binom{M_V}{u+j} \binom{v}{u} \lambda_V^{u+j} (1-\lambda_V)^{M_V-u-j} \mu_V^u (1-\mu_V)^{v-u}. \end{aligned} \tag{30}$$

Finally, for transitions to a smaller number of voice calls, we have

$$\begin{aligned} Pr(V_{k+1} = V_k - j \mid V_k = v) &= Pr(A_{k-1}^V = U_k - j \mid V_k = v) \\ &= \sum_{u=j}^{v} \binom{M_V}{u-j} \binom{v}{u} \lambda_V^{u-j} (1-\lambda_V)^{M_V-u+j} \mu_V^u (1-\mu_V)^{v-u}. \end{aligned} \tag{31}$$

Comments on the Iterative Process

The iteration proceeds in a manner similar to that used for the IFFO protocols, which was discussed in Section 4. The only significant difference is that each stage of the iteration is now a two-step procedure that consists of a data transition followed by a voice transition. Computational issues related to the iterative process are discussed in Appendix B.

9. EVALUATION OF DATA-PACKET DELAY UNDER PR-VD-NIFFL

In this section we discuss the evaluation of expected data-packet delay under the PR-VD-NIFFL protocol. Both fixed- and movable-boundary models are considered. The derivation is similar in principle to that for the IFFO protocols [4, 13], but is somewhat more difficult, primarily because of the absence of the "flush-out" feature. This characteristic, which guarantees that packets arriving in frame k are transmitted by the end of frame $k + 2$, greatly facilitates the delay analysis of IFFO. In the movable-boundary case the analysis is further complicated by the need to model the stochastic nature of the number of slots available for data-packet transmission.

Our objective is to determine the expected delay (or "system time") of data packets, which we define to be the expected time elapsed between the time a packet arrives at a terminal until it is successfully delivered to its destination. In the remainder of this report, we use these two terms interchangeably. The expected delay of a packet depends not only on the number of other packets that are generated during the same frame but also on the system state (R_k, V_k) at the time of its arrival. This is because the current backlog of data packets (R_k) must be transmitted before the new arrivals can be transmitted, and because the statistics of the number of data slots available in future frames (under the movable-boundary version) depend on V_k. Our approach, for both fixed- and movable-boundary schemes, is to determine the conditional expected system time, given the state (R_k, V_k), and then to average this over the probability distribution of the state.

We assume that the system is in state (R_k, V_k) in frame k. We are interested in the A_{k-1} data packets that arrived in frame $k - 1$, since the slots available to serve them depend (deterministically in the fixed-boundary case and stochastically in the movable-boundary case) on (R_k, V_k). We want to determine the expected delay of the A_{k-1} packets that arrived in frame $k-1$, for a given state (R_k, V_k), for all values of A_{k-1}. (Recall that reservations for the A_{k-1} new arrivals are transmitted during the status slot of frame k, and that these packets are transmitted during frame $k+1$ or later.) We define

$$T(A_{k-1} \mid R_k, V_k) = \text{expected total system time of the } A_{k-1} \text{ arrivals, given } (R_k, V_k).$$

$$= \sum_{i=0}^{A_{k-1}} t(i \mid R_k, V_k), \tag{32}$$

where $t(i \mid R_k, V_k)$ is the conditional expected system time experienced by the ith packet, given the state (R_k, V_k). This must be evaluated for every value of A_{k-1}.

The expected delay per packet can then be expressed as

$$E(D) = \sum_{R_k=0}^{\infty} \sum_{V_k=0}^{V_{max}} E(D \mid R_k, V_k)\, Pr(R_k, V_k), \tag{33}$$

where the equilibrium distribution $Pr(R_k, V_k)$ is determined by executing the iterative procedure discussed earlier in this report. The conditional expected total delay, given the state in frame k, is

$$E(D \mid R_k, V_k) = \frac{\sum_{i=1}^{\infty} T(A_{k-1} \mid R_k, V_k)\, Pr(A_{k-1} = i)}{E(A)}. \tag{34}$$

where

$$Pr(A_{k-1} = i) = \binom{ML}{i} \lambda^i (1 - \lambda)^{ML-i} \tag{35}$$

and

$$E(A) = E(A_{k-1}) = E(A_k) = M\lambda L, \tag{36}$$

the expected number of data-packet arrivals per frame, is independent of the state (R_k, V_k) and frame number.

Fixed-Boundary Scheme

Under the fixed-boundary PR-VD-NIFFL scheme, slots numbered $V_{max} + 2$ through L of each frame are available for data, independently of the voice process. Slots in the voice compartment that are not needed for voice calls simply remain idle, even when data packets are waiting in queue. Thus all conditional distributions depend only on R_k and are independent of V_k.

The A_{k-1} new arrivals are not transmitted until the backlog of R_k packets (for which reservations have already been received by the beginning of frame k) are transmitted. At most $(L - V_{max} - 1)$ data packets can be transmitted in frame k. The remaining

$$\tilde{R}_k = \max\left\{[R_k - (L - V_{max} - 1)],\ 0\right\} \tag{37}$$

excess packets are postponed to later frames. Therefore, we have

$$R_{k+1} = \tilde{R}_k + A_{k-1}. \tag{38}$$

If there are no excess packets (i.e., if $\tilde{R}_k = 0$), the first of the A_{k-1} packets is transmitted in the first data slot (slot $V_{max} + 2$) of frame $k+1$. If there are excess packets, the first of the A_{k-1} packets is transmitted in the first data slot after the last of the $\tilde{R}_k$ excess packets has been transmitted. For small values of $\tilde{R}_k$ (i.e., $\le L - V_{max} - 2$) the first slot will be available in frame $k + 1$; for larger values of $\tilde{R}_k$, the first slot will not be available until a later frame. Therefore, the first of the A_{k-1} data packets is transmitted in the $(\tilde{R}_k + 1)$st data slot that occurs after the beginning of frame $k + 1$. Clearly, the last of the A_{k-1} arrivals will be transmitted in the (R_{k+1})st data slot that occurs after the beginning of frame $k + 1$.

Figure 7 shows the relationships among the quantities relating to slot numbers and to the number of data packets that are to be transmitted. We consider the case of $L = 12$ slots and $V_{max} = 6$, in which case there are five data slots in every frame. Three frames are shown: k, $k + 1$, and $k + 2$. The slots are numbered in several different ways. First, for computing delay, all slots (including the status slot, voice slots, and data slots) are numbered consecutively, beginning with the first slot of frame $k + 1$. Next, we consider the slots that are available for data, and number them beginning with the first data slot of frame $k+1$. This permits the evaluation of the delay associated with the ith data packet. These two numberings are independent of any particular realization of the data-packet process. Now, let us consider an example in which $R_k = 8$. Five of these packets are transmitted in the data slots of frame k, and the remaining $\tilde{R}_k = 3$ excess packets are transmitted in the first three data slots of frame $k + 1$. The first of the A_{k-1} new arrivals is transmitted in the fourth data slot of frame $k + 1$, and every data slot in every frame thereafter is available for these A_{k-1} arrivals until the last of them has been transmitted.

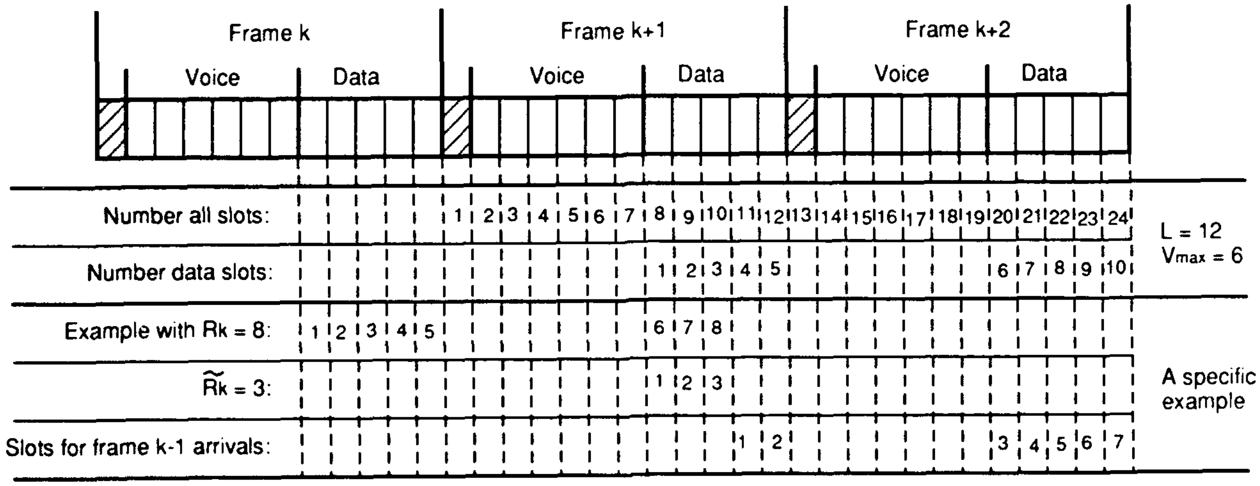


Fig. 7 — Numbering of slots for evaluation of delay under fixed-boundary PR-VD-NIFFL

Because the $\tilde{R}_k$ excess packets are transmitted before the A_{k-1} arrivals, the total system time experienced by the A_{k-1} arrivals is equal to the total system time of all R_{k+1} packets, minus that of the $\tilde{R}_k$ excess packets. We define $\tilde{T}(R_{k+1})$ to be the total system time of R_{k+1} data packets, the first of which is transmitted in the first data slot of frame $k + 1$. Then, the total system time of the A_{k-1} arrivals, which are delayed by the $\tilde{R}_k$ excess packets, can be written as

$$T(A_{k-1} \mid R_k) = \tilde{T}(R_{k+1}) - \tilde{T}(\tilde{R}_k). \tag{39}$$

Our approach is to evaluate $\tilde{T}(R_{k+1})$, which permits the evaluation of $T(A_{k-1} \mid R_k)$ in this manner. Note that $\tilde{T}(R_{k+1})$ is independent of both V_k and R_k because, under the fixed-boundary scheme, $L - 1 - V_{max}$ slots are available for data in every frame.

We first consider the delay that each of the A_{k-1} packets experiences prior to frame $k + 1$. Since the average arrival time of these packets is the midpoint of frame $k - 1$, each packet experiences an average delay of $L/2$ slots in frame $k - 1$. Similarly, because each packet is present in the system throughout frame k, it experiences a deterministic delay of L slots in frame k. Therefore, each packet experiences an average delay of $3L/2$ slots prior to frame $k + 1$. In the following discussion, all additional delays are measured from the beginning of frame $k + 1$. We also note that, once each packet is transmitted, it experiences an additional deterministic delay equal to the round-trip propagation delay of R slots before being received at its destination, where, as noted earlier, $L \geq R$.

Each frame contains $(L - V_{max} - 1)$ data slots. The total number of R_{k+1} packets can be written in terms of the number of frames needed to transmit them as

$$R_{k+1} = K(L - V_{max} - 1) + R_{k+1}^{mod}. \tag{40}$$

where

$$K = \left\lfloor \frac{R_{k+1}}{L - V_{max} - 1} \right\rfloor \tag{41}$$

is the number of complete frames needed to transmit the N packets, and

$$R_{k+1}^{mod} = R_{k+1} \text{ modulo } (L - V_{max} - 1) \tag{42}$$

is the number of packets that must be transmitted in the incomplete frame, which is numbered $K + 1$.

In each of the K complete frames, the first packet is transmitted in slot $V_{max} + 2$ and the last is transmitted in slot L. We consider first the total system time experienced by packets that are transmitted in frame $k + 1$. Because the delays of the successive data packets transmitted in the same frame form an arithmetic series, the total system time of these $L - V_{max} - 1$ packets is easily seen to be

$$\frac{L - V_{max} - 1}{2} \left((V_{max} + 2) + L \right) . \tag{43}$$

Similarly, in frame $k + 2$, since each transmitted packet experiences a delay exactly L slots greater than that of the corresponding packet in frame $k + 1$, the combined system time of the $(L - V_{max} - 1)$ packets is

$$\frac{L - V_{max} - 1}{2} \left((L + V_{max} + 2) + 2L \right) = \frac{L - V_{max} - 1}{2} \left(V_{max} + 2 + 3L \right) . \tag{44}$$

In general, the combined total delay of the $L - V_{max} - 1$ packets transmitted in frame $k + m$ is

$$\frac{L - V_{max} - 1}{2} \left([(m-1)L + V_{max} + 2] + mL \right)$$

$$= \frac{L - V_{max} - 1}{2} \left(V_{max} + 2 + (2m-1)L \right) . \tag{45}$$

Summing the contribution from the K complete frames yields

$$\frac{L - V_{max} - 1}{2} \sum_{m=1}^{K} \left(V_{max} + 2 + (2m-1)L \right)$$

$$= \frac{L - V_{max} - 1}{2} \left[K(V_{max} + 2) + L \sum_{m=1}^{K} (2m - 1) \right]$$

$$= \frac{L - V_{max} - 1}{2} \left[K(V_{max} + 2) + LK^2 \right] . \tag{46}$$

Now we address the contribution to total system time associated with the R_{K+1}^{mod} packets transmitted in frame $K + 1$. The combined delay of these packets is

$$\frac{R_{K+1}^{mod}}{2} \left((KL + V_{max} + 2) + (KL + V_{max} + 1 + R_{K+1}^{mod}) \right)$$

$$= \frac{R_{K+1}^{mod}}{2} \left(2KL + 2V_{max} + 3 + R_{K+1}^{mod} \right) . \tag{47}$$

Combining the results from the complete frames and the incomplete frame, we have

$$\bar{T}(R_{k+1}) = \frac{L - V_{max} - 1}{2} \left[K(V_{max} + 2) + LK^2 \right]$$

$$+ \frac{R_{K+1}^{mod}}{2} \left(2KL + 2V_{max} + 3 + R_{K+1}^{mod} \right) . \tag{48}$$

This expression is used in conjunction with Eq. (39) to determine $T(A_{k-1} \mid R_k)$. It is then possible to determine $E(D \mid R_k)$ by using Eq. (34), and finally $E(D)$ by using Eq. (32). Because system operation under the fixed-boundary scheme is independent of the voice process, conditioning on the voice component of the state V_k is not needed. Recall that the expression for $T(R_{k+1})$ does not include the expected delay experienced during frame $k - 1$ (an average of $L/2$ slots per packet) and frame k (exactly L slots per packet), nor does it include the round-trip propagation delay once the packet is actually transmitted (exactly R slots per packet). Thus a delay of $3L/2 + R$ slots must be added to the value computed for expected system time to incorporate these delays. Therefore, we obtain for the conditional expected system time per packet:

$$E(D \mid R_k) = \frac{\sum_{i=1}^{ML} \left[\tilde{T}(\tilde{R}_k + i) - \tilde{T}(\tilde{R}_k) \right] \binom{ML}{i} \lambda^i (1 - \lambda)^{ML - i}}{M \lambda L} + \frac{3L}{2} + R , \tag{49}$$

where $\tilde{R}_k$ is uniquely determined by R_k for given values of L and V_{max} by using Eq. (37), and $\tilde{T}(R_{k+1})$ is given by Eq. (48). The overall expected delay is

$$E(D) = \sum_{j=0}^{\infty} E(D \mid R_k) \, Pr(R_k = j). \tag{50}$$

where $Pr(R_k)$ is the equilibrium distribution of R_k, which is obtained by following the iterative procedure for the data-only PR-NIFFL protocol.

Clearly, the summations in the above expressions must be truncated at some finite value, as was done in the evaluation of the equilibrium distribution of the state probability $Pr(R_k, V_k)$. Otherwise, the evaluation of expected system time is exact.

Movable-Boundary Scheme

Under the movable-boundary PR-VD-NIFFL scheme, the set of slots available for data packets in each frame depends on the voice process, which is a random process. Unused slots in the voice compartment are available for data-packet transmission. For example, in frame $k + m$, since V_{k+m} ($\le V_{max}$) voice calls are in progress, $L - V_{k+m} - 1$ slots are available for data-packet transmission. Thus the number of excess packets is now

$$\tilde{R}_k = \max \left\{ [R_k - (L - V_k - 1)], 0 \right\} . \tag{51}$$

Note that $\tilde{R}_k$ now depends on V_k, whereas for the fixed-boundary scheme V_{max} was used for all values of V_k. The evolution of the voice-call process (and hence the number of slots available for data on a frame-by-frame basis) is defined by the Markov chain with transition probabilities $Pr(V_{k+1} \mid V_k)$, which are independent of the data process.

The stochastic nature of the sequence of slots available for data transmission has prevented us from finding closed-form expressions for $\tilde{T}(R_{k+1})$. However, we have developed a procedural method for accurately evaluating this quantity. In the following discussion, we illustrate the source of difficulty in pursuing a purely analytical approach and provide a complete description of the procedural approach.

Another feature of this system that complicates the analysis is that the V_k voice calls in progress during frame k do not necessarily occupy the first V_k slots of the voice compartment; a voice call retains its particular slot throughout the call duration, and each call is equally likely to terminate in any given frame. Thus data packets can be inserted in empty slots and intermingled among the voice packets. Although it would be possible to keep track of which slots are occupied by voice calls, to do so would require an extremely complicated model. Instead, we make the simplifying assumption that the V_k voice calls do occupy the first V_k slots in the voice compartment (i.e., slots 2 through V_k + 1). This provides an upper bound on the expected system time for data packets that is accurate to within a fraction of a frame.

Following the analysis for the fixed-boundary case, we see that the combined total expected system time of the $L - V_{k+m} - 1$ packets transmitted in frame $k + m$ (again not including the common term of $3L/2 + R$) is

$$\frac{L - V_{k+m} - 1}{2}\left(V_{k+m} + 2 + (2m - 1)L\right) . \tag{52}$$

For any given sequence of the voice process $\{V_k, V_{k+1}, \cdots, V_{k+m}, \cdots\}$, and for any total number of packets ($R_{k+1} = R_k + A_{k-1}$), it is possible to determine the number of complete frames needed (K) and the number of data slots in frame $K + 1$ (i.e., R_{k+1}^{mod}). The total system time of the R_{k+1} packets is

$$\begin{aligned}\tilde{T}(R_{k+1}) &= \frac{1}{2}\sum_{m=1}^{K}(L - V_{k+m} - 1)\left(V_{k+m} + 2 + (2m-1)L\right) \\ &+ \frac{R_{k+1}^{mod}}{2}\left(2mL + 2V_{k+K+1} + 3 + R_{k+1}^{mod}\right) .\end{aligned} \tag{53}$$

Although this quantity can be computed easily for any particular sequence, this approach is not practical, except for small numbers of packets, because of the number of possible sequences that can occur. For example, in the course of K frames, there are $(V_{max} + 1)^K$ possible sequences. Furthermore, the number of frames needed depends on the particular sequence. To avoid the rapid increase in the size of the state space that occurs as R_{k+1} increases, and the resulting computational difficulties, we have taken an approach that exploits the Markovian nature of the voice-call process.

In the analysis of the fixed-boundary scheme, we were able to exploit the fact that the sequence of time slots available for data traffic is deterministic, thus permitting the exact determination of the slot number in which the ith data packet is transmitted. In the movable-boundary case, however, because the number of slots available each frame is a random process, the slot number in which any particular packet is transmitted is a random variable. Thus, instead of the exact slot number, we can evaluate

$$t(i \mid R_k, V_k) = E(\text{slot in which packet } i \text{ is transmitted} \mid R_k, V_k), \quad 1 \le i \le A_{k-1}. \tag{54}$$

This, in turn, permits us to evaluate $T(A_{k-1} \mid R_k, V_k)$ for all values of A_{k-1} and then the expected system time averaged over the system state. We again have

$$T(A_{k-1} \mid R_k) = \tilde{T}(R_{k+1}) - \tilde{T}(\tilde{R}_k). \tag{55}$$

To simplify the notation we set $k = 0$, so that frame $k + m$ is now designated as frame m, and the number of arrivals in frame $k - 1$ is now designated as A_{-1}. We consider the transition from frame m to frame $m + 1$. We introduce a new state* description $(D_{(m-1)L+j}, V_m)$, where $D_{(m-1)L+j} = d$ if the dth data packet (of the total number R_{k+1}, numbered cumulatively from frame 1) is transmitted in the jth slot of frame m, and 0 if no data packet is transmitted in that slot.

Given the probability distribution of the state $(D_{(m-1)L+j}, V_m)$, the probability that packet d is transmitted in slot $mL + j$ is easily determined as follows:

$$Pr(D_{mL+j} = d \mid V_0 = v_0) = \sum_{v=0}^{V_{max}} Pr(D_{mL+j} = d, V_{m+1} = v \mid V_0 = v_0), \quad 2 \le j \le L. \tag{56}$$

Once these probabilities are determined, we can define

$$\tilde{t}(d \mid V_0 = v_0) = E(\text{slot number in which packet } d \text{ is transmitted} \mid V_0 = v_0)$$

$$= \sum_{j=1}^{\infty} j \, Pr(D_j = d \mid V_0 = v_0). \tag{57}$$

Here we are interested in the dth packet of the combined total of $R_1 = \tilde{R}_0 + A_{-1}$ packets. To determine the corresponding total system time of the dth packet, we simply add the common delay of $3L/2 + R$ to Eq. (57). The expected total system time of the A_{-1} arrivals, given the initial state (R_0, V_0), can then be evaluated as

$$T(A_{-1} \mid R_0, V_0) = \sum_{d=R_0+1}^{R_1} \tilde{t}(d \mid V_0) + \left(\frac{3L}{2} + R \right) A_{-1} . \tag{58}$$

The expected delay per packet is then determined by using Eqs. (33) and (34).

Determination of State Probabilities: $Pr(D_{(m-1)L+j}, V_m)$

To determine the state probabilities, we first observe that no data slots are available for any of the $R_1 = \tilde{R}_0 + A_{-1}$ packets of interest during frame 0. Thus we have

$$Pr(D_R = 0, V_0 = v_0) = 1 \tag{59}$$

for the specified value of $V_0 = v_0$. Starting from this initial condition, we iteratively determine the state probabilities for each slot in successive frames. The procedure described below is repeated for all values of V_0 between 0 and V_{max}. Knowledge of the state probability in the last slot of frame m (i.e., (D_{mL}, V_m)) is sufficient to determine the state probabilities in all slots of frame $m + 1$. Note that as long as $V_{max} < L - 1$, a condition that would be satisfied in any practical integrated system, at least the last slot of every frame will always be available for data. Therefore, we have an imbedded first-order Markov process, which is defined at the last slot of every frame.

First we consider transitions to states in which $V_{m+1} = 0$. The transitions with nonzero probability are summarized by:

$$\left.\begin{array}{r} Pr(D_{mL+2} = d + 1, V_{m+1} = 0) \\ Pr(D_{mL+3} = d + 2, V_{m+1} = 0) \\ \cdots \\ Pr(D_{(m+1)L} = d + L - 1, V_{m+1} = 0) \end{array}\right\} = \sum_{V_m=0}^{V_{max}} Pr(D_{mL} = d, V_m) \, Pr(V_{m+1} = 0 \mid V_m). \tag{60}$$

*The use of the term "state" to describe the pair $(D_{(m-1)L+j}, V_m)$ is confined to this section; elsewhere in this report we use the term state to refer to the pair (R_k, V_k).

Figure 8 shows these transitions for a simple example in which $L = 8$ and $V_{max} = 3$. We assume that the state in the last slot of frame m is $(D_{mL}, V_m) = (d, v)$. If $V_{m+1} = 0$ (an event that occurs with probability $Pr(V_{m+1} = 0 \mid V_m = v)$), slots 2 through L of frame $m+1$ are available for data. Thus the $d+1$st packet is transmitted in the second slot (the first data slot) of frame $m+1$, the $d+2$nd packet is transmitted in the third slot, etc. Note that the first line below the timing diagram of Fig. 8 represents the case for transitions to $V_{m+1} = 0$. Transitions to other values of V_{m+1}, which are discussed below, are shown on the following lines.

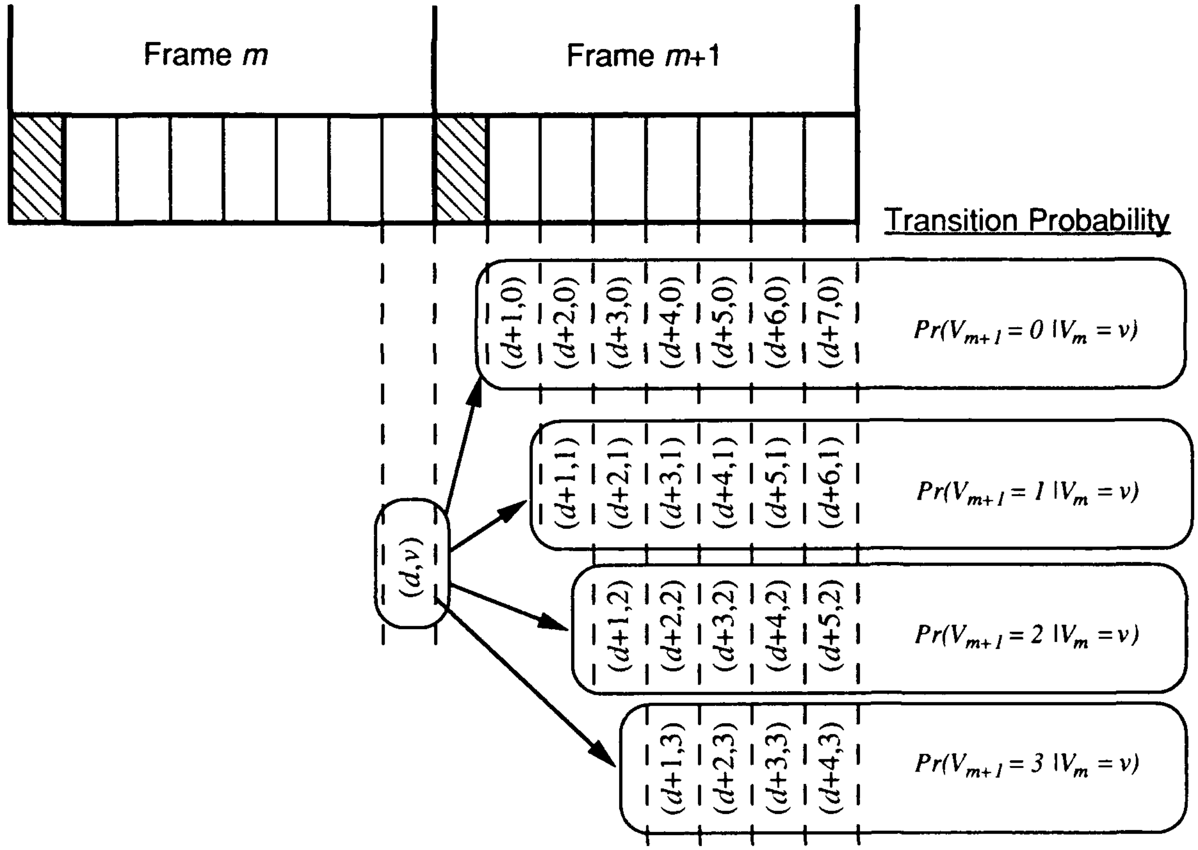


Fig. 8 — Evolution of the state (d, v) under movable-boundary PR-VD-NIFFL

These calculations are performed for all values of d such that $Pr(D_{mL} = d, V_m)$ is nonzero. Note that all of the transitions listed in Eq. (60) have the same probability because they each are a consequence of a voice transition to a state in which $V_{m+1} = 0$, in which case $L-1$ data slots are available in frame $m+1$. The conditional probability of this transition, given V_m, is independent of D_{mL}. Whenever $V_{m+1} = 0$, $L-1$ data slots are available in frame $m+1$. Thus data packets are transmitted in slots 2 through L, and the total number of data packets served increases by $L-1$ over the duration of the frame. Note that the summation in Eq. (60) includes transitions from all possible values of V_m to $V_{m+1} = 0$, whereas Fig. 8 represents the transitions from a particular value of V_m.

In general, for $0 \le V_{mL} = v \le V_{max}$, in which case data packets are transmitted in slots $v+2$ through L,

$$\left.\begin{array}{r} Pr(D_{mL+2+v} = d+1, V_{m+1} = v) \\ Pr(D_{mL+3+v} = d+2, V_{m+1} = v) \\ \cdots \\ Pr(D_{(m+1)L} = d+L-(v+1), V_{m+1} = v) \end{array}\right\} = \sum_{V_m=0}^{V_{max}} Pr(D_{mL} = d, V_m)\, Pr(V_{m+1} = v \mid V_m). \quad (61)$$

Computational issues associated with the evaluation of data-packet delay for the movable-boundary scheme are discussed in Appendix B.

Performance results obtained by using this method are presented in Section 11. Here we remark that our results have verified that the expected system time computed by using this model does, in fact, approach that of the more-easily-computed fixed-boundary model in the limit of extremely high voice traffic (in which case no voice slots are available for data, so the data-packet delay performance under the movable-boundary scheme approaches that of a fixed-boundary one).

10. EVALUATION OF VOICE-CALL BLOCKING PROBABILITY

The performance measure used for voice traffic is the voice-call blocking probability P_B. Under all of the integrated protocols considered in this report, voice calls are accepted by the system as long as doing so does not cause the number of calls in progress simultaneously to exceed V_{max}. Whenever a slot is not available for a new call, the call is blocked and dropped from the system. Thus P_B depends only on V_{max} and the voice-call arrival (λ_V) and completion (μ_V) probabilities, and is independent of the data-traffic parameters. Therefore, P_B is the same for all of the integrated protocols considered in this report. We define

B_k = number of calls blocked in frame k.

The blocking probability is then

$$P_B = \frac{E(B_k)}{E(A_k^V)}, \tag{62}$$

where the expectation is taken over the equilibrium distribution of the voice-call process V_k.

We first consider the conditional distribution of the number of blocked voice calls, given V_k and U_k (the number of voice-call completions in frame k):

$$Pr(B_k = j \mid V_k = v, U_k = u) = Pr(A_{k-1}^V = V_{max} - (V_k - U_k) + j \mid V_k = v, U_k = u)$$

$$= \binom{M_V}{V_{max} - v + u + j} \lambda_V^{V_{max} - v + u + j} (1 - \lambda_V)^{M_V - (V_{max} - v + u + j)}. \tag{63}$$

Here we have used the fact that the distribution of A_k^V is independent of V_k because the arrival process at each voice terminal is independent of the number of voice calls already active at that terminal. It is a Bernoulli trial with probability λ_V at every voice terminal in every frame. Averaging over the conditional distribution of U_k, given V_k, yields

$$Pr(B_k = j \mid V_k = v) = \sum_{u=0}^{v} Pr(B_k = j \mid V_k = v, U_k = u)\, Pr(U_k = u \mid V_k = v), \tag{64}$$

where

$$Pr(U_k = u \mid V_k = v) = \binom{v}{u} \mu_V^u (1 - \mu_V)^{v-u}. \tag{65}$$

The conditional expectation of B_k, given V_k, is then

$$E(B_k \mid V_k) = \sum_{j=1}^{M_V} j\, Pr(B_k = j \mid V_k = v). \tag{66}$$

Combining the above results yields

$$E(B_k \mid V_k) = \sum_{j=1}^{M_V} j \sum_{u=0}^{v} \binom{M_V}{V_{max} - v + u + j} \lambda_V^{V_{max} - v + u + j}$$

$$\times (1 - \lambda_V)^{M_V - (V_{max} - v + u + j)} \binom{v}{u} \mu_V^u (1 - \mu_V)^{v - u}. \tag{67}$$

We then average over the distribution of V_k to obtain

$$E(B_k) = \sum_{v=0}^{V_{max}} E(B_k \mid V_k)\, Pr(V_k = v), \tag{68}$$

where the equilibrium distribution for V_k, obtained by means of the iterative procedure described in Section 8, is used in this equation.

The only remaining quantity needed to evaluate P_B by using Eq. (62) is the expected number of voice arrivals per frame. To determine this quantity we again make use of the fact that the arrival process is independent of the system state to obtain

$$E(A_k^V) = M_V \lambda_V, \tag{69}$$

which results in

$$P_B = \frac{E(B_k)}{M_V \lambda_V}. \tag{70}$$

11. PERFORMANCE RESULTS

Both performance measures discussed in this report, namely data-packet delay and voice-call blocking probability, have been evaluated computationally. In this section we present the results of our evaluation of both fixed- and movable-boundary versions of the PR-VD-NIFFL protocol. All of the results are for $M = 10$. Our earlier studies of the IFFO protocols have shown that performance is quite insensitive to the number of terminals for M greater than about 5 and this behavior has also been observed for the PR-VD-NIFFL protocol; thus the results for $M = 10$ are representative of higher values as well. Unless otherwise noted, our results are for $L = R = 12$. The number of iterations required for convergence ranged from 2 at extremely low throughput rates to more than 2000 at extremely high throughput rates.

Data Traffic

Figure 9 shows the expected data-packet system time (which is normalized with respect to the frame length L) as a function of data throughput for the fixed-boundary PR-VD-NIFFL scheme for values of V_{max} ranging from 0 to 6. Note that data throughput is defined to be the expected number of data packets successfully delivered by the system per time slot, where the average is taken over all L slots in the frame, including the status slot and those that are used for voice transmission. Recall that this performance index is independent of all voice-call parameters except V_{max} in fixed-boundary systems. Note that the curve for $V_{max} = 0$ is actually the curve for PR-NIFFL, which is a data-only system. At low to moderate throughput rates, the expected delay increases linearly with throughput because not many of the packets that arrive in frame k are delayed past frame $k + 2$. The maximum achievable data throughput under the PR-VD-NIFFL schemes is $(L - 1 - V_{max})/L$, and we may define the "utilization" to be the data throughput multiplied by $L/(L - 1 - V_{max})$. Asymptotes are

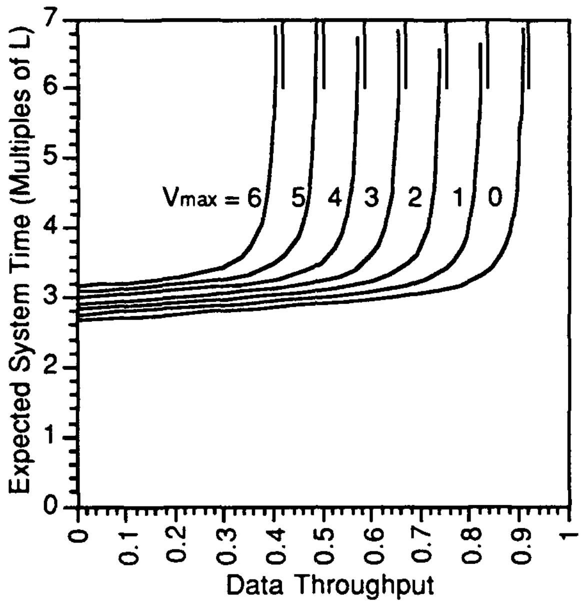


Fig. 9 — Normalized expected system time for data packets under fixed-boundary PR-VD-NIFFL as a function of data throughput (L = 12, M = 10)

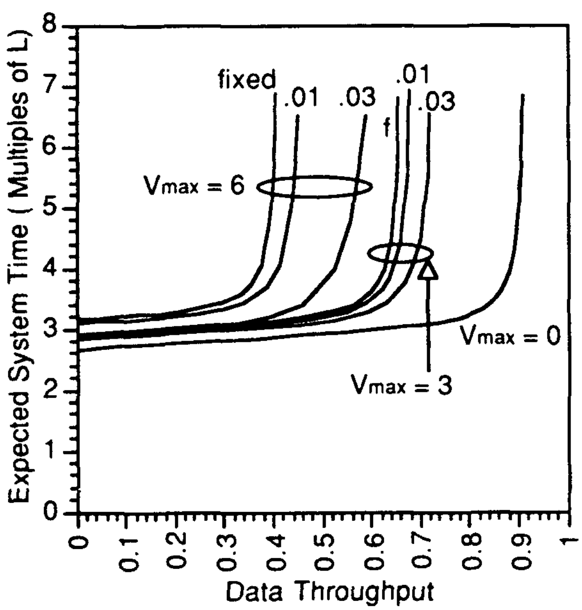


Fig. 10 — Normalized expected system time for data packets under fixed- and movable-boundary PR-VD-NIFFL as a function of data throughput (L = 12, M = 10, λ_V = 0.01)

drawn to show these maximum throughput values, at which point the utilization is equal to 1 and the expected system time increases to ∞. For all values of V_{max}, results were obtained for utilization values of at least 0.98 (for which a transition matrix of size 700 $\times$ 700 was used), thereby demonstrating the ability of our model to evaluate heavily loaded systems. As V_{max} increases, less of the channel resource is available for data traffic; therefore, the curves begin their ascent at lower values of data throughput.

Figure 10 shows a comparison of the normalized expected system time for the fixed- and movable-boundary versions of PR-VD-NIFFL for V_{max} = 0, 3, and 6. For all curves shown, M_V = 10 and λ_V = 0.01. For V_{max} = 3 and 6, curves are shown for μ_V = 0.01 and 0.03; since the case of V_{max} = 0 represents a data-only system (i.e., PR-NIFFL), only a single curve is shown for it. For μ_V = 0.01, in which case the voice compartment is fairly heavily loaded, the movable-boundary scheme provides only slightly better performance than the fixed-boundary scheme. When μ_V = 0.03, the average length of each voice call is decreased by a factor of 3 as compared to the previous case. For the case of V_{max} = 6, the movable-boundary scheme provides significant improvement over the fixed-boundary scheme because a considerable number of slots in the voice compartment are now used by data packets. The improvement is less pronounced for the case of V_{max} = 3, since in that case the voice compartment is still heavily loaded, and thus not many slots are available for data.

Voice Traffic

Figure 11 shows the effect of varying μ_V (and hence the expected length of voice calls) over a wide range for the movable-boundary scheme with V_{max} = 6, while λ_V is kept fixed at 0.01. For small values of μ_V, the system is heavily loaded, and performance approaches that of the fixed-boundary scheme. As μ_V increases, the average length of voice calls decreases. This results in a decrease in the voice-call load and hence an increase in the number of slots available for data traffic. Thus the system is able to support higher levels of data traffic. The rightmost curve, i.e., that for V_{max} = 0, represents the limiting case in which the entire channel resource is available for data traffic.

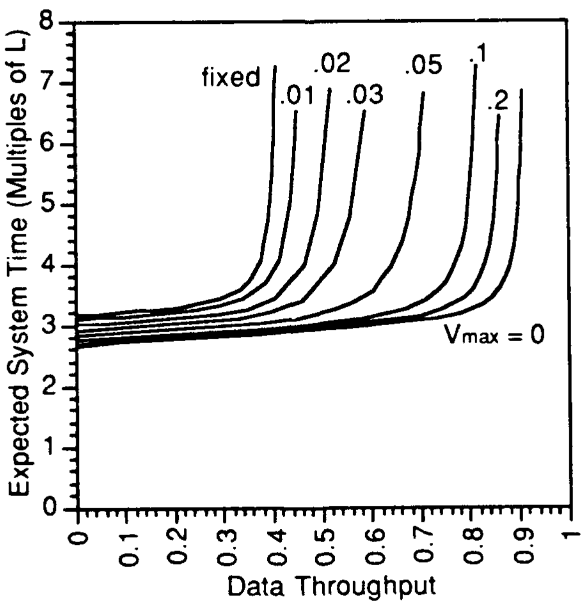


Fig. 11 — Normalized expected system time for data packets under movable-boundary PR-VD-NIFFL as a function of data throughput, showing dependence on μ_V (L = 12, M = 10, V_{max} = 6, λ_V = 0.01)

Figure 12 shows the expected number of active voice calls as a function of V_{max} for $M_V = 10$, $\lambda_V = 0.01$, and several values of μ_V. For small values of μ_V (long call durations), the system is heavily loaded and the number of voice calls increases in almost direct proportion to V_{max}. For large values of μ_V, the number of voice calls is limited primarily by the available traffic rather than by the channel resource.

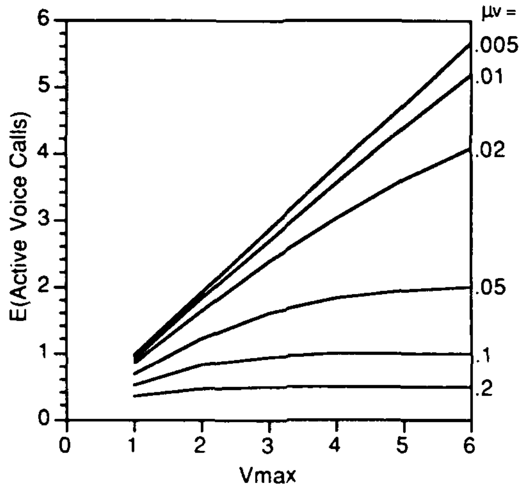


Fig. 12 — Expected number of active voice calls under PR-VD-NIFFL as a function of V_{max} (M_V = 10, λ_V = 0.01)

Figure 13 shows the voice-call blocking probability as a function of V_{max} for $M_V = 10$, $\lambda_V = 0.01$, and several values of μ_V. Clearly, increasing V_{max} results in lower blocking probability because more of the channel resource is available for the voice calls. Also, since large values of μ_V correspond to short voice calls, increasing μ_V results in lower channel utilization and hence in lower blocking probability. Note that the expected number of active voice calls and the voice-call blocking probability are independent of the data parameters and process, and are the same for fixed- and movable-boundary versions.

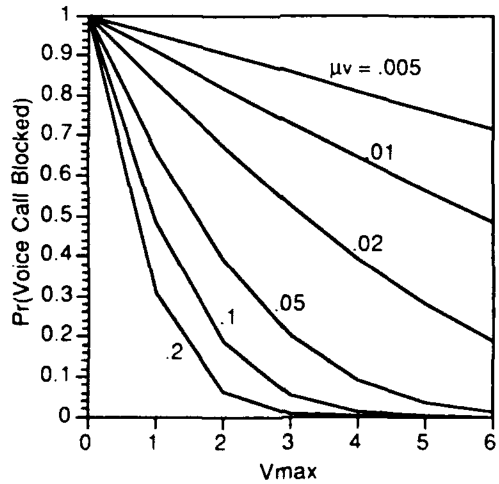


Fig. 13 — Voice-call blocking probability under PR-VD-NIFFL as a function of V_{max} (M_V = 10, λ_V = 0.01)

Combined Performance Measure

It is clear that the desire to minimize data-packet delay and to minimize voice-call blocking probability represents conflicting goals. The former is minimized when $V_{max} = 0$, whereas the latter is minimized when V_{max} is equal to its maximum permitted value, which, in the examples considered thus far, is 6. In such cases with two competing performance indexes, it is customary to attempt to minimize their weighted sum. Thus we would like to minimize

$$E(D) + \alpha P_B.$$

Figure 14 shows this weighted performance index as a function of V_{max} for the fixed-boundary version and two values of α, i.e., 2 and 8, where delay is again normalized with respect to L. In each case, curves are plotted for a fixed value of data-packet throughput. Note that for throughput values of 0.48 and greater, the curves terminate at values of $V_{max} < 6$; in each of these cases the value of throughput corresponds to a utilization of 0.96 for the corresponding value of V_{max}. Throughput values that correspond (for a specific value of V_{max}) to a utilization of 1.0 or greater result in infinite delay, and hence an infinite value of the weighted performance index.

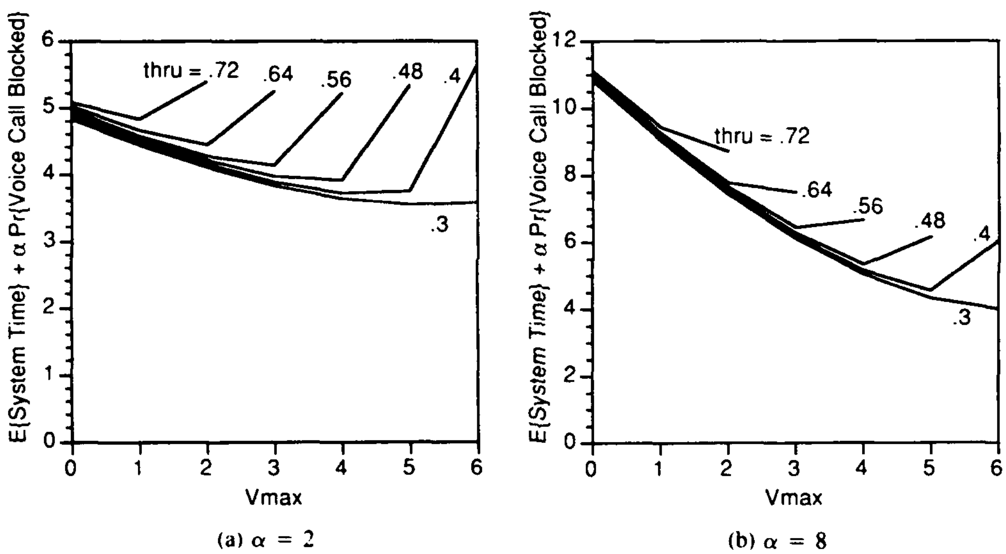


Fig. 14 — Weighted performance index for fixed-boundary PR-VD-NIFFL (L = 12, M = 10)

For $\alpha = 0$, only the expected system time is of concern, since no importance is given to blocking probability; thus the best performance is obtained for $V_{max} = 0$. For $\alpha = 2$, the trade-offs between the two performance indexes are apparent; as throughput decreases the optimal value of V_{max} increases. For $\alpha = 8$, more importance is assigned to voice-call blocking probability, and higher values of V_{max} than for the previous case are appropriate for some of the higher values of throughput. For significantly higher values of α, it is best to use the highest value of V_{max} that will support the required data-packet throughput.

Data Packet Delay for Different Frame Lengths

All of the results presented thus far have been for the case of $L = 12$. Figure 15 shows the normalized expected system time for a fixed-boundary system for the case of $L = 120$ and $V_{max} = 0$, 30, 60, and 90. In each of these curves, the maximum value of utilization for which results are presented is 0.98. Figure 16 shows the normalized expected system time as a function of utilization

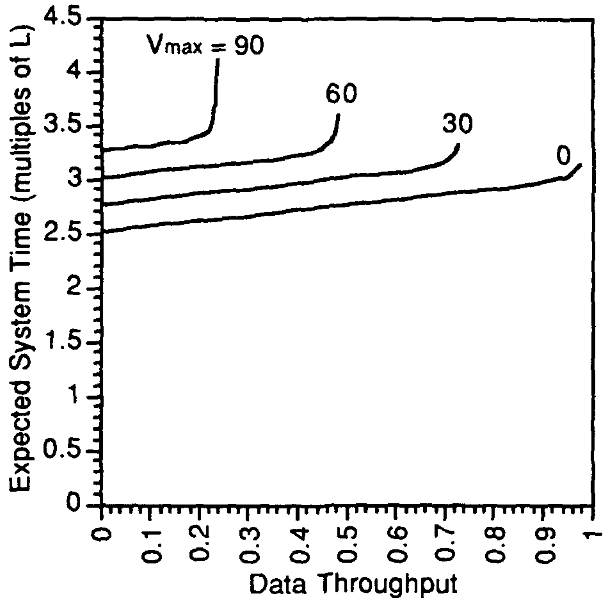


Fig. 15 — Normalized expected system time for data packets under fixed-boundary PR-VD-NIFFL as a function of data throughput (L = 120, M = 10)

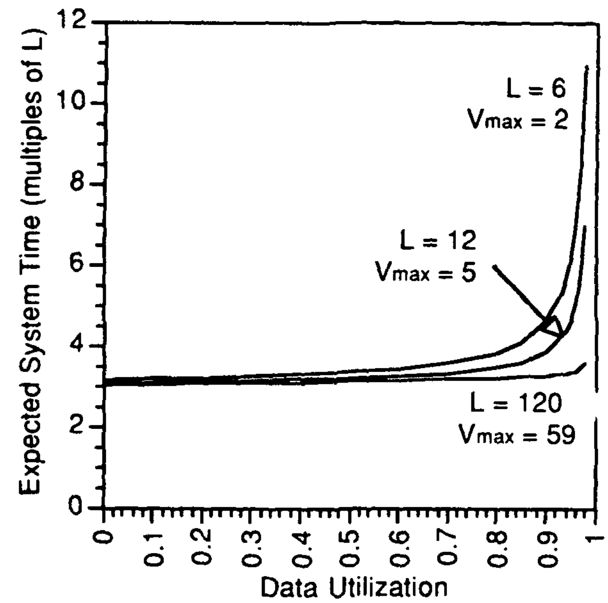


Fig. 16 — Normalized expected system time for data packets under fixed-boundary PR-VD-NIFFL as a function of data utilization for three values of L (M = 10)

for $L = 6$, 12, and 120. Curves are shown for values of V_{max} such that half of the slots in each frame are available for data, i.e., $V_{max} = L/2 - 1$; the maximum utilization value is 0.98 for each case. This figure shows that, for similar utilization levels and a similar fraction of the frame length dedicated to data transmission, the normalized expected system time is considerably smaller for large values of L than for small values. Furthermore, the curves start their ascent from the linear region at higher values of utilization when L is large.

This effect is even more pronounced for examples with larger values of V_{max}, since a smaller portion of the frame is available for data traffic. The superior performance at greater frame lengths* apparently results from the impact of the law of large numbers, i.e., the ratio of the standard deviation of the number of arrivals per frame to the expected number of arrivals decreases as frame length increases. This behavior is similar to that of statistical multiplexers, since only high-capacity systems of that type can work at near-capacity levels without losing many packets.

12. OPERATION IN A GROUND-RADIO ENVIRONMENT

Geostationary satellite channels are characterized by a round-trip propagation delay of approximately 0.27 second, which may typically correspond to values of R that may range from 10 to 1200, depending on data rate and packet size, as discussed in Section 4. In contrast, in ground radio channels the propagation delays are typically measured in ms, which correspond to $R << 1$ in low-data-rate systems such as those commonly used in military applications (although we may have $R > 1$ in high-data-rate systems). When $R << 1$, we may assume that reservations are received instantaneously (since guard times would be incorporated into each slot anyway). As in the case of $R > 1$, reservations for packets that arrive in frame k are sent in the first slot of frame $k + 1$. However, since the reservation information is available immediately, slots for these packets can be reserved in frame $k+1$ instead of waiting until frame $k+2$. In this case, the interleaved frame structure of IFFO and IFFL is not needed. For example, in Ref. 13 a Pure-Reservation Direct-Flush-Out (PR-DFO) scheme was discussed, and simple exact expressions were derived for expected system delay. Similarly, when considering fixed frame lengths there is no need to incorporate the intermediate frames associated with IFFL. The effectively instantaneous reception of reservations permits the NIFFL schemes (including the hybrid schemes that use the unreserved slots for transmission on a contention basis) to be described by a first-order Markov chain that represents transitions from R_k to R_{k+1}.† In particular, the complex second-order Markov chain description needed to characterize F-NIFFL for the case of $R > 1$ is no longer needed because the transition probabilities for the NIFFL schemes become identical to those for the IFFL schemes when the propagation delay is negligible.

When $R << 1$, there is normally no reason to incorporate contention operation into the IFFO protocol because reservations can be made in the next unreserved slot since the frame length can shrink to as small as one slot (i.e., only the status slot) when traffic is light. However, when considering integrated voice/data systems, it is appropriate to maintain a fixed frame length of L slots because each packet of a voice call needs a periodically recurring time slot. Thus it may be appropriate to maintain frame lengths of at least some specified length even when the propagation delay is near zero as is done under IFFL and NIFFL.

The delay analysis of PR-VD-NIFFL schemes in which the propagation delay is much less than the frame length follows directly from that given in this report. The only differences are that there is no longer an intermediate frame (since packets arriving in frame k can now be transmitted in frame

*Here we refer to normalized delay. Of course, the absolute delay increases as L increases.

†In integrated voice/data protocols, data transmissions also depend on V_k.

$k + 1$ instead of frame $k + 2$) and that the propagation delay associated with the actual data packet is now negligible. Thus, instead of the common term $3L/2 + R$ found in the delay expressions of Eqs. (49) and (58), we would obtain $L/2 + R$ since the delay of R slots associated with frame $k+1$ is no longer incurred.

13. CONCLUSIONS

In this report we have addressed the major issues associated with channel access in integrated wireless networks, and we have proposed and analyzed the Pure-Reservation Voice/Data Non-Interleaved-Frame Fixed-Length (PR-VD-NIFFL) protocols. These schemes are especially well suited to satellite networks and to high-speed wireless terrestrial networks in which the round-trip propagation delay is greater than the data-packet duration, although they are also appropriate for environments in which propagation delay is less than a packet duration, a common characteristic of military environments. The techniques we have developed to exploit the structural properties of the two-dimensional first-order Markov chain that characterizes this scheme have been shown to permit the efficient and accurate evaluation of data-packet delay and voice-call blocking probability. Performance results illustrate the dependence of performance on system parameters, and demonstrate the improved performance that can be achieved through the use of the movable-boundary version as compared to the fixed-boundary version. Trade-offs between data-packet delay and voice-call blocking probability show the importance of the parameter V_{max}, the threshold used to control access of voice calls to the system.

REFERENCES

1. F.A. Tobagi, "Multiaccess Protocols in Packet Communication Systems," *IEEE Trans. Comm.*, **COM-28**, 468-488 (1980).

2. S.S. Lam, "Multiple Access Protocols," in *Computer Communications, Volume I: Principles*, ed. W. Chou (Prentice Hall, 1983).

3. J.E. Wieselthier and A. Ephremides, "A New Class of Protocols for Multiple Access in Satellite Networks," *IEEE Trans. Automatic Control* **AC-25**, 865-879 (1980).

4. J.E. Wieselthier, "A New Class of Multi-Access Protocols for Packet Communication Over a Satellite Channel—The Interleaved Frame Flush-Out Protocols," Ph.D. dissertation, University of Maryland (March 1979).

5. L.G. Roberts, "Dynamic Allocation of Satellite Capacity Through Packet Reservation," *AFIPS Conference Proceedings, 1973 National Computer Conference* **42** (1973) pp. 711-716.

6. G.J. Coviello, and P.A. Vena, "Integration of Circuit/Packet Switching by a SENET (Slotted Envelope Network) Concept," *Conference Record of National Telecommunications Conference* (1975) pp. 42.12-42.17.

7. K. Sriram, P. Varshney, and J.G. Shanthikumar, "Discrete-Time Analysis of Integrated Voice/Data Multiplexers With and Without Speech Activity Detectors," *IEEE J. Selected Areas Commun.* **SAC-1**, 1124-1132 (1983).

8. T. Suda, H. Miyahara, and T. Hasegawa, "Performance Evaluation of an Integrated Access Scheme in a Satellite Communication Channel," *IEEE J. Selected Areas Commun.* **SAC-1**, 153-164 (1983).

9. C.-S. Wu, and V.O.K. Li, "Integrated Voice/Data Protocols for Satellite Channels," *Conference Record of Mobile Satellite Conference*, Jet Propulsion Laboratory (1988) pp. 413-422.

10. M. Soroushnejad and E. Geraniotis, "Voice/Data Integration in Random-Access Code Division Multiple-Access Packet Radio Networks," *Proceedings of the Twenty-Third Annual Conference on Information Sciences and Systems (CISS)*, Johns Hopkins University (March 1989) pp. 706-710.

11. E. Geraniotis and M. Soroushnejad, "Performance Evaluation of Multi-Access Strategies for an Integrated Voice/Data CDMA Packet Radio Network," NRL Memorandum Report 6743 (November 1990).

12. L.G. Roberts, "ALOHA Packet System With and Without Slots and Capture," ARPANET Satellite System Note 8 (NIC 11290), June 1972; reprinted in *Comput. Commun. Rev.* **5**, (1975).

13. J.E. Wieselthier and A. Ephremides, "Protocols of Multiple Access (A Survey) and the IFFO Protocols (A Case Study)," *Proceedings of the NATO Advanced Study Institute on "New Concepts in Multi-User Communication,"* Norwich, U.K. 1980, (Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 1981).

14. F.G. Foster, "On the Stochastic Matrices Associated with Certain Queueing Processes," *Ann. Math. Statist.* **24**, 355-360, (1953).

15. J.E. Wieselthier and A. Ephremides, "Some Markov Chain Problems in the Evaluation of Multiple-Access Protocols," in *Numerical Solution of Markov Chains*, ed. W.J. Stewart (1991) pp. 239-26 (Marcel Dekker, Inc., New York).

16. J.E. Wieselthier and A. Ephremides, "A Movable-Boundary Channel-Access Scheme for Integrated Voice/Data Networks," *Proceedings of INFOCOM'91* (April 1991) pp. 721-731.

17. J.E. Wieselthier and A. Ephremides, "Movable-Boundary Channel-Access Schemes for Integrated Voice/Data Networks," NRL Memorandum Report 6706 (August 1990).

18. I. Viniotis and A. Ephremides, "Optimal Switching of Voice and Data at a Network Node," *Proceedings of the 28th Conference on Decision and Control (CDC)* (December 1987) pp. 1504-1507.

19. I. Viniotis and A. Ephremides, "On the Optimal Dynamic Switching of Voice and Data in Communication Networks," *Proceedings of the Computer Networking Symposium* (April 1988) pp. 8-16.

Appendix A
HYBRID VERSIONS OF THE IFFO, IFFL, AND NIFFL SCHEMES

In the body of this report we limited our discussion primarily to pure-reservation versions of IFFO, IFFL, and NIFFL. In this appendix we demonstrate how the analysis is extended to hybrid versions of these data-only schemes in which the fixed-contention mode of channel access, defined in Section 4, is used in the unreserved slots (if any) of each frame. These hybrid schemes are designated as F-IFFO, F-IFFL, and F-NIFFL.

In Ref. B1 we demonstrated that F-IFFO provides considerable improvement over PR-IFFO at low to moderate throughput rates. Similar improvement is expected under F-IFFL and F-NIFFL, as compared with PR-IFFL and PR-NIFFL, although performance results for the hybrid versions have not yet been obtained. It would be straightforward to implement a F-VD-NIFFL scheme; system operation would be the same as that under PF-VD-NIFFL, except that F-NIFFL would be used in place of PR-NIFFL in the data compartment of each frame.

We now discuss the hybrid versions of the protocols we have studied.

F-IFFO

For F-IFFO we can easily show that (see Ref. A1)

$$p_{ij} \triangleq Pr(R_{k+2} = j \mid R_k = i) = \binom{M(i+1)}{j} \lambda^j (1-\lambda)^{M(i+1)-j} * c_{N_k}(j),$$

where * represents the convolution operator and

$$c_{N_k}(j) = c(j) * c(j) * \cdots * c(j)$$

is the convolution of N_k (which depends on $R_k = i$) pmf's of the form $c(j)$, which is the probability that j packets are transmitted unsuccessfully in a contention slot. All convolutions are performed numerically. In particular,

$$c(0) = (1-\lambda)^{M-1}[(1-\lambda) + M\lambda]$$

$$c(1) = 0$$

$$c(j) = \binom{M}{j} \lambda^j (1-\lambda)^{M-j}, \quad 2 \le j \le M.$$

Thus $c_{N_k}(j)$ is the pmf of the number of unsuccessful (colliding) packets in a frame with N_k contention slots. The expression to the left of the convolution operator in the transition probability for F-IFFO is the pmf of the number of arrivals in frame k that do not attempt transmission in contention slots (because of the slot number in which they arrived; see the discussion on transmission policy in

Section 4). Whenever $R_k \geq R-1$, the convolution vanishes because $N_k = 0$; thus the p_{ij} are the same as those for PR-IFFO. Therefore, the performance of F-IFFO is closely bounded by that of PR-IFFO in the limit of high input rates.

F-IFFL

Under F-IFFL the frame length is fixed at L slots, where $L \geq R$. It is easy to observe that for $R_k < L-1$ the transition probabilities derived for F-IFFO apply, whereas for $R_k \geq L-1$ those for PR-IFFL must be used; hence the transition probabilities for F-IFFL are:

$$p_{ij} \triangleq Pr(R_{k+2} = j \mid R_k = i) = \begin{cases} \binom{M(i+1)}{j} \lambda^j (1-\lambda)^{M(i+1)-j} * c_{N_k}(j), & 0 \leq i \leq L-1 \\ \binom{ML}{j-i+L-1} \lambda^{j-i+L-1} (1-\lambda)^{ML-(j-i+L-1)}, & i \geq L-1. \end{cases}$$

F-NIFFL

In Section 6 we demonstrated that PR-NIFFL can be described by a first-order Markov chain. Unfortunately, it is not possible to do so for F-NIFFL. However, the second-order Markov chain can be decomposed to generate a collection of smaller problems, as we now discuss. We start by observing that R_{k+1} is common to both states (i.e., origin and destination) of each transition. This led to the state probability description in terms of the vectors of the form $\mathbf{q}_j$ with elements q_{ji}; the index j takes on all possible values of R_{k+1} (i.e., from 0 to $N-1$) and i takes on all possible values of R_k (i.e., from 0 to $L-1$).* For each value of R_{k+1} we consider the transitions from R_k to R_{k+2}. The corresponding transition probabilities are denoted as

$$p_{i \frac{j}{} m} \triangleq p_{ij-jm} \triangleq Pr(R_{k+2} = m,\ R_{k+1} = j \mid R_{k+1} = j,\ R_k = i).$$

Whenever $j \leq L-1$, $R_{k+2}^{(k+1)} = 0$, in which case $R_{k+2} = R_{k+2}^{(k)}$. Whenever $j > L-1$, $R_{k+2}^{(k+1)} = j-(L-1)$ packets must be added to $R_{k+2}^{(k)}$. By an appropriate modification of the expressions previously derived for F-IFFL, we have the following transition probabilities for F-NIFFL:

$$p_{i \frac{j}{} m} = \begin{cases} \binom{M(i+1)}{m} \lambda^m (1-\lambda)^{M(i+1)-m} * c_{N_k}(m), & 0 \leq j \leq L-1 \\ \binom{M(i+1)}{m-j+L-1} \lambda^{m-j+L-1} (1-\lambda)^{M(i+1)-(m-j+L-1)} * c_{N_k}(m), & j > L-1. \end{cases}$$

Recall that we must have $i \leq L-1$ since $L-1$ is an aggregate state (the excess packets are incorporated into $j = R_{k+1}$). Again, the convolution vanishes whenever $i = L-1$. Note that the two expressions given here correspond to different ranges of j, rather than i. There are N transition probability matrices of this type (one for each value of j, denoted $\mathbf{P}_{(j)}$), each of dimension $L \times N$. Each $\mathbf{q}_j$ vector is multiplied by the corresponding transition probability matrix $\mathbf{P}_{(j)}$. The result is again a collection of N probability vectors for R_{k+2}, each of dimension N, one for each value of R_{k+1}. The vector corresponding to $R_{k+1} = j$, which we denote $\mathbf{r}_j$, consists of the elements

$$r_{jm} = Pr(R_{k+1} = j,\ R_{k+2} = m).\dagger$$

*As noted in Section 6, the state $R_k = L-1$ is an aggregate state that actually contains all values of $R_k \geq L-1$

†Note that $\mathbf{q}_j$ refers to a probability vector in which the state in the later of two slots is held constant (i.e., $R_{k+1} = j$, while R_k varies), whereas $\mathbf{r}_j$ refers to a probability vector in which the state in the earlier of two slots is held constant (i.e., $R_{k+1} = j$, while R_{k+2} varies)

In preparation for the next iteration, the state probabilities are rearranged in the form of the $\mathbf{q}_j$ vectors, where all states for which $R_{k+1} \geq L-1$ are combined in the aggregate state $L-1$. This procedure is repeated until convergence is achieved.

In our discussion of the IFFO protocols we made the observation that the performance of PR-IFFO bounds closely that of F-IFFO in the limit of high throughput rates. Similarly, at high throughputs the performance of PR-NIFFL provides a tight upper bound on that of F-NIFFL. This is true because most frames have no unreserved slots at high throughput rates, in which case $R_k = L-1$ (the aggregate state). Thus no k-packets can be transmitted in frame k, in which case F-NIFFL functions in the same manner as PR-NIFFL. Therefore, at high throughput rates (for which large transition probability matrices are needed), the performance of F-NIFFL can be bounded and closely approximated by that of PR-NIFFL, which is characterized by a much simpler description (i.e., a first-order Markov chain).

REFERENCE

A1. J.E. Wieselthier and A. Ephremides, "A New Class of Protocols for Multiple Access in Satellite Networks," *IEEE Trans. Automatic Control* **AC-25**, 865-879 (1980).

Appendix B
COMPUTATIONAL ISSUES

It is computationally intensive to evaluate the performance of the protocols studied in this paper. In this appendix, we discuss some of the major issues we have encountered and several of the techniques we have used to ease the computational burden.

Truncation of the Transition Probability Matrix

The infinite Markov chain that describes R_k has been truncated to a finite dimension N. An $N \times N$ transition probability matrix permits the modeling of up to $N - 1$ reserved slots per frame (since the N elements in the probability vector range from 0 to $N - 1$). The truncation operation involves setting $Pr(j = N - 1 \mid i) = 1 - Pr(j \leq N - 2 \mid i)$ to maintain conservation of probability. Typical values of N can range from 20 at low throughput to 700 at extremely high throughput for the IFFO, IFFL, and NIFFL protocols. The largest value of N used for the integrated PR-VD-NIFFL scheme has been reduced to 250 because of the need to store $V_{max} + 1$ of these transition probability matrices (i.e., up to seven of them in our examples). In our studies we have verified that the values of N used in our calculations have been sufficiently large so that the effect of truncation error is small.

Combinatorial Expressions

The evaluation of the combinatorial expressions of the form $\binom{M(i+1)}{j}$, which appear in the transition probability for the IFFO protocols, raises an interesting computational problem. For large values of M (values up to 50 have been considered) and i (as great as 700) the values of the combinatorial expressions become extremely large, while the values of the corresponding exponential expressions become extremely small. To avoid the resulting computational problems, we recognize that, since the arrival process in each slot is an independent sequence, the pmf of the number of arrivals in $i+1$ slots is simply the convolution of $i+1$ pmf's of the number of arrivals in a single slot. Thus we can perform the computation by convolving expressions that have less extreme values. We have

$$a(j) = Pr(j \text{ arrivals in one slot}) = \binom{M}{j} \lambda^j (1 - \lambda)^{M-j}.$$

Therefore, for $i \geq R - 1$,

$$p_{ij} = a(j) * a(j) * \cdots * a(j),$$

where * represents the convolution operator and the expression is the convolution of $i + 1$ pmf's of the form $a(j)$. Similarly, for $i < R - 1$, p_{ij} is the convolution of R $a(j)$'s. All convolutions are performed numerically.

Termination Criterion

In the computations for the data-only protocols, the iteration was stopped when the following criterion was satisfied:

$$|\pi_j(n+1) - \pi_j(n)| \leq 0.5 \times 10^{-6}, \qquad 0 \leq j \leq N-1,$$

i.e., each element in the pmf changes by not more than 0.5×10^{-6} from one iteration to the next. In our numerical evaluation of PR-VD-NIFFL, the iteration was stopped when each element of the joint pmf of (R_k, V_k) changed by no more than 10^{-7} from one iteration to the next. To verify that this criterion is sufficiently stringent, several runs at high throughput values were made with a tightened criterion of 10^{-8}. This resulted in an increase of the evaluated data-packet delay of at most several tenths of one percent.

The number of iterations needed to satisfy this criterion ranged from two at extremely low throughput to more than 2000 at extremely high throughput. We have observed that, as the frame length L increases while utilization and the ratio $(V_{max}+1)/L$ remain essentially constant, the number of iterations required to achieve convergence decreases significantly, as does the normalized expected system time; this was discussed in Section 11.

A scaling operation, which ensures that the elements of the pmf sum to 1 at each iteration, must be included to minimize the effects of computer roundoff error.

Initial State

In all of our numerical examples for the data-only protocols (IFFO, IFFL, and NIFFL), we assumed arbitrarily that $\pi(0)$ is uniformly distributed between 0 and 9; no attempt was made to optimize the initial state. For the integrated PR-VD-NIFFL scheme, we again start from an initial state in which the data-packet component of the state is uniformly distributed over this interval. However, the initial voice state is not uniformly distributed. We exploit the fact that, as a consequence of the simple threshold policy for accepting voice calls (*i.e., accept a call as long as doing so would not cause the number of active voice calls to exceed the threshold* V_{max}), voice traffic is independent of data traffic. Thus the equilibrium distribution of V_k, which is the same for all versions of the VD-IFFL and VD-NIFFL protocols, can be determined independently of that of the data traffic. This property has been exploited in establishing the initial condition from which the iteration is started. In particular, we have set

$$Pr(R_0 = r, V_0 = v) = 0.1\, Pr(V_k = v), \qquad 0 \leq r \leq 9,\ 0 \leq v \leq V_{max}.$$

Use of this initial condition has reduced the required number of iterations considerably, as compared to the use of a uniform distribution over the voice states. For example, for a case with low data throughput, convergence was achieved in three iterations (as compared to 182 for a uniform initial distribution) with identical performance results.

Evaluation of Delay for the Movable-Boundary Scheme

A three-dimensional array is needed to store the state probabilities of the form $Pr(D_{(m-1)L+j} = d, V_m = v)$, i.e., for the dimensions of slot number $((m-1)L + j)$, index of the data packet that is transmitted (d), and number of voice calls in the current frame (V_m). For example, for a frame length of $L = 12$ slots, $V_{max} = 6$, and if we wish to keep track of up to $d = 250$ packets, an array of size $(L-1) \times (V_{max}+1) \times N = 11 \times 7 \times 250 = 19{,}250$ per frame is needed.* The entries of this array are filled in one frame at a time by using the iterative procedure described in Section 9. For the specified system parameters, up to 50 frames may be needed to transmit 250 data packets if voice traffic is heavy, thus requiring the storage of almost 1 million

*Note that $L - 1$ (rather than L) appears here because the first slot of every frame (the status slot) is never available for data.

probability values if a straightforward brute-force approach is used. Fortunately, however, storage requirements can be reduced greatly, as shown in the following.

Although the calculation of the expected slot number in which the dth packet is transmitted requires a summation over all slot numbers, we can define partial sums and update them after each frame. For example, to represent the contribution to expected delay of the dth packet that is accumulated in frame $m + 1$, we can define

$$\tilde{t}_{m+1}(d \mid V_0 = v_0) = \tilde{t}_m(d \mid V_0 = v_0) + \sum_{j=mL+1}^{(m+1)L} j \, Pr(D_j = d \mid V_0 = v_0),$$

where

$$\tilde{t}_0(d \mid V_0 = v_0) = 0.$$

Thus, to evaluate the system time, it is necessary to store the three-dimensional probability array for only one frame at a time. Another helpful feature is the fact that, to determine the transitions from frame m to frame $m + 1$, only those entries corresponding to slot mL (the last slot of frame m) are needed. Finally, we observe that the state probabilities of the form $Pr(D_{(m-1)L+j}, V_m)$, which characterize the slots available for data packets, are independent of the data parameters. Thus it is possible to perform most of the delay-related calculations independently of the calculations of the equilibrium system state (R_k, V_k). For example, if the same set of voice-call parameters is used for several sets of data-packet parameters, the same set of $Pr(D_{(m-1)L+j}, V_m)$ values can be used for each of them.