We extend the \( \pi \)-persistent protocol proposed in [1] to include voice for multiaccess communication over unidirectional broadcast bus networks. We employ a framed approach for integrating these two traffic types. Further, we not only use speech detectors for modeling speech sources, but we also employ a movable-boundary scheme for our protocol in which the voice subframe size is estimated from the number of voice packets transmitted in the previous frame. We determine the fraction of speech loss and the fraction of wasted allocated bandwidth due to the estimator by formulating a Markov chain for the number of ready voice stations at the frame boundaries. To analyze data performance, we construct a nonhomogeneous Markov chain for a data station's buffer content just before the data slots visit that station. Then, we convert these models to homogeneous chains from which we determine the optimum \( \pi \) for each data slot. We find that, by proper control of the estimator function, we can guarantee an upper bound on the average speech loss at any voice station. Moreover, a combined voice-data throughput close to unity is achieved, limited only by the wasted bandwidth due to the estimator. Simulation results which agree with the performance predicted by our analysis are also included.

I. INTRODUCTION

Stream traffic, such as digitized speech, possesses some bursty properties which render packet switching a feasible alternative for its transmission as opposed to circuit switching [2]. This is supported by numerous proposals for integrating packet voice with data over links [3-5], bidirectional buses [6-9], rings [10-12], unidirectional buses [13-14], and other networks [2]. Via such integration, we not only avoid duplication of resources, but also gain by utilizing the available resources equitably. Further, we can also integrate other traffic types like digitized video and facsimile into the network. Our attention herein will be focused on voice-data integration over unidirectional buses that are long and involve high bandwidth, e.g., fiber-optic bus networks.

Proposed protocols for voice-data integration over unidirectional buses consider either alternate cycles of voice and data [13-15], or several cycles of voice followed by several cycles of data forming a frame which is repeated [16], or repetitive cycles each of which contain voice as well as data packets [17]. In all of these protocols, a cycle is terminated by a gap due to signal propagation delay on the bus. These gaps, which represent wasted bandwidth, result in a capacity loss which increases when the propagation delay normalized to a packet's transmission time increases [18, 19]. To overcome this capacity loss, we propose a new voice-data integration protocol suitable for long, high-speed, unidirectional buses, and, then, we model and analyze its performance capabilities.

The characteristics and transmission requirements of voice traffic are considerably different from those of data traffic. Statistics from speech sources indicate that only about 40% of the time is a speaker actually generating talkspurts during a conversation [20]. Also, voice traffic requires near real-time delivery, although it can tolerate some loss [21]. Data traffic, on the other hand, can tolerate longer, variable delays but very little, if any, loss. Consequently, a framed approach is often employed for integrating these two traffic types where a frame, which is repeated, consists of a voice subframe followed by a data subframe [3, 4].

When no speech detectors are used to suppress the silent periods in speech, the boundary between the two subframes is kept fixed. If speech detectors are used, we can prevent the loss of about 60% of the bandwidth that would otherwise be required, but, then, we also need a movable-boundary protocol [3]. To avoid the capacity loss due to cycles, we employ a movable-boundary scheme for our protocol in which the voice subframe size is estimated from the number of voice packets transmitted in the previous frame. This estimator function has similarities with the allocation function in [4]. Access to the data subframe is provided to the data stations via the \( \pi \)-persistent protocol which was developed in [1].

We describe the voice-data integration protocol and its underlying assumptions in Section II and present its performance analysis in Section III. For voice, we determine the fraction of speech loss and the fraction of wasted allocated bandwidth due to the estimator by formulating a Markov chain for the number of ready voice stations at the frame boundaries. For analyzing data performance, we employ a nonhomogeneous Markov chain for a data station's buffer content just before the data slots visit that station. Then, we obtain the average data packet delay as well as the probability of blocking and the throughput. Numerical results are presented in
Section IV. We find that, by proper control of the estimator function, we can guarantee an upper bound on the average clipping experienced by any voice station. Moreover, a combined voice-data throughput close to unity is achieved, limited only by the wasted bandwidth due to the estimator. We also study the dependence of the delay-throughput characteristics of data traffic on various system parameters. To verify our analysis, we have developed a simulation model of our protocol. We find that there is a close match between the results of our analysis and simulation. We conclude in Section V with a review and discussion of our results.

II. VOICE-DATA INTEGRATION

A. Network Organization

We consider the Single-bus Unidirectional Broadcast System (SUBS) network topology [26] with NV voice stations and Nd data stations as shown in Fig 1(a). We note that our approach can also be applied to the Dual-bus Unidirectional Broadcast System (DUBS) network topology [26] by the use of proper handshaking signals between the end stations as in Fastnet [16, 22]. Because of the unidirectional nature of the transmission medium, we can identify a best-case layout wherein all voice stations access the bus before any of the data stations as shown in Fig. 1(b). It is also possible that all of the data stations access the bus before any of the voice stations in which case we have the worst-case layout as indicated in Fig. 1(c). The general case, in which the order of bus access among the voice and data stations is unknown, will also be referred to as the worst-case layout. This is because any layout which is not the best-case layout will require voice subframe estimation as described below in Section III(C).

We assume that each data station can buffer M packets, 1 ≤ M < ∞, while each voice station can buffer only one packet, the packet length being equal for both traffic types. A voice station's buffer is filled by a voice digitizer according to the following voice station model.

B. Voice Station Model

Using statistical measurements, a speech source has been modeled by the three-state Markov process given in Fig. 2. In the idle state, no conversation takes place, but the source becomes active when conversation starts. In our analysis, we will assume that NV voice sources are continually active. An active source alternates between the talk and silent states. The amount of time spent in each of these states is exponentially distributed with means 1/α and 1/μ, respectively. Experiments [20, 23, 24] have been performed to estimate 1/α and 1/μ, and, for our model, we will use 1/α = 1.5 s and 1/μ = 2.25 s [16], with speech activity being α/μ + α) = 0.4.

Each voice source is equipped with a vocoder that digitizes speech at a rate RV bps. Silent periods in speech are suppressed. A voice station is allowed to transmit only one packet in a frame whose structure is described in the following subsection. The duration of a frame is less than the maximum tolerable speech delay which, for intelligible speech, is around 100 ms. Therefore, for analytical tractability, we modify the above voice station model slightly by assuming that short talkspurts are ignored while short silent periods are "filled-in". In particular, we assume that, over the duration of a frame, a voice station makes at most one transition between its talk and silent states [14, 20].

C. Voice Station Protocol

The most upstream station on a bus, be it a voice or data source, is identified as the head station. The head station generates fixed-sized, empty slots on the bus. It marks a new frame after every F slots, where F = [Rv/Rd] is the frame length (in slots), Rv is the channel bandwidth, and [a] is the largest integer smaller than or equal to a. For our protocol, we require that NV ≤ F. We denote by tν(r) the instant the beginning of frame r, r = 1, 2, ..., visits Voice Station ν, ν = 1, 2, ..., NV. We assume that a speech packet is created at Voice Station ν and transferred into that station's buffer at tν(1), r = 1, 2, .... If the station was in the talk state both at tν(r-1) and tν(r), the amount of speech collected will be just sufficient to fill a packet completely. If it was silent at both of the above instants, no speech packet is created. Otherwise, a partial packet is created which is restored to the regular packet size by filling in its silent period. The packet assembled during frame r-1 is transmitted in frame r.

For the best-case layout, a voice station accesses the earliest empty slot in a frame to transmit its packet if it has any to send. Since NV ≤ F, there is no speech loss (or clipping) due to the protocol.

For the worst-case (or general) layout, the head station marks the first Lr slots, 1 ≤ Lr ≤ NV, in frame r for use by voice packets only. The integer Lr = Lr(Kr-1) is the estimated voice subframe size of frame r given that Kr-1 voice packets were transmitted in frame r-1. We note that the frame length will be a few tens of milliseconds so that buses up to a few hundred kilometers in length will have at most two partial frames existing simultaneously. This ensures that Lr can easily be obtained since Kr-1 can be determined by tracking the inbound channel (see Fig. 1). Of course, it is possible that more than Lr voice stations are ready to transmit packets in frame r. Then, stations beyond the most upstream Lr ready voice stations will lose their chance to transmit in that frame, the result of this packet loss being that the corresponding speech is clipped. It is clear that, due to the unidirectional access scheme, the last voice station experiences the worst clipping. On the other hand, if fewer than Lr voice stations transmit in frame r, there will be several unused voice slots and, hence, a loss of bandwidth. Thus, the specification of Lr involves a tradeoff between clipping and wasted bandwidth which we will analyze in Section III.

We note that the head station must determine Lr based on feedback from the inbound channel, viz., Kr-1. Since all stations have access to the inbound channel, the above operation is easily distributed among all the stations. Also, for our analysis, we will ignore the inter-packet and intra-packet overhead in order to establish theoretical performance limits of our protocol.

D. Data Station Protocol

For the best-case layout, data stations access the empty slots in a frame according to the p-persistent
protocol [1], while for the worst-case layout, they use the same protocol to access the slots not marked exclusively for voice. Now, the $p_i$ are different from those for the pure data case because the Markov chain specifying the buffer content of Data Station $i$, $1 \leq i \leq N_d$, at its data slot boundaries is no longer homogeneous. The analysis to determine the $p_i$ (for fairness) is presented in Section III.

## III. PERFORMANCE ANALYSIS

### A. Voice

The activity on the inbound channel after all stations have accessed the bus is shown in Fig. 3. For the best-case layout, $K_T = L_T$ while for the worst-case layout, $K_T \leq L_T$. Let $B$ be the packet length in bits, and $f_r$ the length of frame $r$ in seconds. Our protocol requires that $f_r = f$ be independent of $r$. Also, we have

$$B = R_f f$$

(1)

A constraint on the choice of $B$ and, hence, on $f$ is the maximum delay tolerable by speech signals. The voice station whose packets suffer the longest waiting (from formation until completion of transmission) is Station $N_v$. The longest value of this waiting time in any frame $r$ is $f$ (when $N_v = F$, and all voice stations transmit in frame $r$). The earliest speech samples in the packet in question were collected at the beginning of frame $r-1$ implying that they had already been delayed by $f$ when frame $r$ started. Thus, the maximum delay suffered by speech under this protocol is $2f$, ignoring the propagation delay between the source and destination stations. Therefore, we require that $f \leq D_{\text{max}}/2$, where $D_{\text{max}}$ is the maximum delay tolerable by speech excluding the propagation delay. Combining this condition with (1), we require that

$$B \leq D_{\text{max}} R_v/2$$

(2)

We denote by $t(r)$ the instant when frame $r$ is generated by the head station. Since the model of Voice Station $v$ (Fig. 2) possesses the memoryless property, that station's behavior at the instants $t_v(r)$ and $t(r)$ are equivalent. Hence, if we let $n_v(r)$ be the number of ready voice stations at $t(r)$, $n_v(r)$ is also the number of voice stations that will have packets for transmission in frame $r$. Thus, we can use $t(r)$ as a global start-of-frame instant instead of $\{t_v(r), v = 1, 2, \ldots, N_v\}$.

We assume that all voice stations are independent, and note that $n_v(r)$ depends only upon $n_v(r-1)$ and the activities during frame $r-1$, viz., transitions between the talk and the silent states for any of the voice stations. Hence, the sequence $\{n_v(r), r = 1, 2, \ldots\}$ forms an imbedded Markov chain.

The transition probability matrix of this chain is given by [25]

$$v_{jk} = \sum_{w=0}^{\min(j,k)} \left( \begin{array}{c} j \\ w \end{array} \right) \left( \begin{array}{c} N_v - j \\ k-w \end{array} \right) \cdot x_{11}^w \cdot x_{10}^{j-w} \cdot x_{01}^w \cdot x_{00}^{N_v-j-k+w}$$

(3)

where $j, k = 0, 1, 2, \ldots, N_v$. $x_{00} = \exp(-f_0)$ is the probability that a station did not transmit packets in frames $r-1$ and $r$. $x_{01} = 1 - x_{00}$, $x_{11} = \exp(-f_0)$ is the probability that a station transmitted packets in frames $r-1$ and $r$, and $x_{10} = 1 - x_{11}$. Knowing (3), the steady-state probability distribution of the number of ready voice stations \{$\pi_v(V), v = 0, 1, 2, \ldots, N_v$\} follows. However, since the voice stations are independent, $\pi_v(V)$ can also be determined by

$$\pi_v(V) = \left( \begin{array}{c} N_v \\ v \end{array} \right) \cdot \left( 1 - x \right)^{N_v-v}$$

(4)

where $v = 0, 1, 2, \ldots, N_v$, and $x = \mu/\mu + \alpha$ is the probability that a voice station is in the talk state at any arbitrary instant. We use (4) to determine $\pi_v(V)$ because it is direct.

For the best-case layout, there is no clipping since, in any frame, all voice stations transmit their packets before any data stations. Also, since $L_T = K_T$, there is no waste of voice slots.

For the worst-case layout, $L_T$ is estimated by

$$L_T = \min (K_T-1 + \delta, N_v)$$

(5)

where $\delta$ is a nonnegative integer. The estimator in (5) implies that, by knowing the number of packets transmitted in frame $r-1$, we allocate $\delta$ additional slots in order to hold clipping at an acceptable level. The integer $\delta$ is a design parameter whose choice determines the tradeoff between clipping and wasted bandwidth. In (5), we also ensure that we do not allocate more slots for voice than there are voice stations.

We note that $K_T-1$ can be an integer value between $0$ and $N_v$. Letting $K_T-1 = n$, $n = 0, 1, 2, \ldots, N_v$, in (5), we find that $L_T$ is independent of $r$, but depends on $n$. Thus,

$$J_n = \text{min} \left( n + \delta, N_v \right)$$

(6)

Since $J_n$ slots are allocated for voice in frame $r$, Voice Station $N_v$ is clipped when more than $J_n$ voice packets are ready for transmission in frame $r$. The more upstream voice stations, of course, experience less clipping. If fewer than $J_n$ packets are transmitted in frame $r$, there is wastage of slots. Hence, for any frame $r$, we have

$$\Pr[\text{Station } N_v \text{ clipped } | K_T-1 = n] = \sum_{k=J_n+1}^{N_v} v_{nk}$$

(7)

and

$$E[\text{slots wasted } | K_T-1 = n] = \sum_{k=0}^{J_n-1} (J_n-k) v_{nk}$$

(8)

where $J_n$ is given by (6). From (8),

$$E[\text{fraction of wasted allocated bandwidth } | K_T-1 = n] = \frac{\sum_{k=0}^{J_n-1} J_n-k}{J_n} v_{nk}$$

(9)
Removing the conditioning on n in (7) and (9), we get

\[ \Pr[\text{Station } N_v \text{ clipped}] = \frac{N_v}{\sum_{n=0}^{J_n-1} \sum_{k=J_n+1}^{J_n-1} v_{nk} p_n(V)} \]  \hspace{1cm} (10)

and the fraction of wasted allocated bandwidth \( W_b \) is

\[ W_b = \frac{N_v}{\sum_{n=0}^{J_n-1} \sum_{k=0}^{J_n-1} v_{nk} p_n(V)} \]  \hspace{1cm} (11)

Instead of (5), other functions to estimate \( L_r \) can also be employed, e.g.,

\[ L_r = \min\{ E[K_r | K_{r-1} = n] + \delta \} \]  \hspace{1cm} (12)

where \( E[\cdot] \) denotes conditional expectation. The estimator in (12) is apparently more appealing since, by knowing the number of packets transmitted in frame \( r-1 \), it determines the expected number of packets to be transmitted in frame \( r \) and then allocates \( \delta \) additional slots for \( L_r \). Under steady-state conditions, when \( r \to \infty \), the conditional expectation in (12) is approximated by

\[ \lim_{r \to \infty} E[K_r | K_{r-1} = n] = \frac{N_v}{\sum_{k=0}^{J_n-1} v_{nk}} \]  \hspace{1cm} (13)

where the \( v_{nk} \) are given by (3). We remark that the above result is not exact since it is obtained by assuming that there is no clipping in frame \( r-1 \). In other words, we assumed that, at \( (r-1) \), exactly \( n \) voice stations were ready while several more could have actually been ready. Consequently, we choose (5) rather than (12) for determining \( L_r \), but discuss the performance under both estimators in Section IV.

B. Data

We let data slots be those slots which are not used by voice for the best-case layout or not marked for voice for the worst-case layout. We define the instants at which data slots visit Data Station \( i \) to be imbedded times for that station. The imbedded times are shown in Fig. 4 in which we also show consecutive frames each with a voice subframe of \( J_n \) slots and a data subframe of \( F-J_n \) slots.

Since the frame length is a few tens of milliseconds, \( \chi_0 \) and \( \chi_1 \) in (3) are close to unity which implies that \( \{v_{jk}\} \) is almost diagonal. Hence, \( E[K_r | K_{r-1} = n] = n \) and \( J_n = n + \delta \) has a variance close to zero. This allows us to use the frame structure in Fig. 4 to evaluate data performance by assuming that \( J_n \) is initially a constant. Subsequently, by allowing \( J_n \) (or, alternately, \( n \)) to take on all possible values, we obtain the overall data performance.

We assume that the traffic originating at Data Station \( i \) is Poisson with rate \( \lambda_i \). By also assuming that the traffic distribution is symmetric, we get \( \lambda_i = \lambda N_d \) where \( \lambda \) is the (Poisson) data traffic rate offered to the entire network. We recall from [1] that, in SUBS, all stations receive identical service and that all three fairness criteria, viz., equal-delay, equal-blocking, and equal-throughput, are equivalent. Hence, we consider only the performance of Data Station \( i \) under the equal-delay policy.

We denote Data Station \( i \)'s imbedded times in a frame by \( \tau \) where \( \tau = 1, 2, \ldots, F-J_n \) as shown in Fig. 4. We note that this station's buffer content at imbedded time \( \tau \) depends on the buffer content at the previous imbedded time, viz., \( (\tau-2)+F-J_n \mod(F-J_n) + 1 \), and on any activities between these two times, viz., packet arrivals, or a packet transmission, or neither. Hence, Data Station \( i \)'s buffer contents at these imbedded times forms a Markov chain. This chain, however, is nonhomogeneous since the transition from \( \tau = F-J_n \) to \( \tau = 1 \) is different from the other transitions. Nevertheless, by grouping every \( F-J_n \) consecutive transitions into one, we can construct homogeneous chains which can then be solved.

To this end, we let \( G_i(\tau) \) represent the transition between \( \tau \) and \( \tau \mod(F-J_n) + 1 \) for Data Station \( i \) and note from [1] that, for \( M > 1 \), its elements are given by

\[ G_i(\tau) = \begin{bmatrix} 0 & 2j \leq M, 0 \leq k < j - 1 \\ u_i(\tau)^{c_{0j}} & 1 \leq j < M, k = j - 1 \\ c_ki & j = 0, 0 \leq j < M - 1 \\ u_i(\tau)^{c_{kj}} & 1 \leq k < j - 1, 0 \leq j < M - 1 \\ (1-u_i(\tau))^{c_{kj}} & 1 \leq k < j - 1, 0 < j < M - 1 \\ 1-d_Mi & j = M \end{bmatrix} \]  \hspace{1cm} (14)

where \( c_{ai} = \Pr[a \text{ arrivals in } \mu \text{ slots}] = (\lambda \mu)^a \exp(-\lambda \mu) / a! \) where \( a = 0, 1, 2, \ldots, b \) and \( d_{bi} = 1 - \sum_{a=0}^{b} c_{ai} \). We note that this is the same as (2) in [1] except that the probabilities \( u_i, \tau_i, \) and \( p_i \) are now functions of \( \tau \). For \( M = 1 \), we get

\[ G_i(\tau) = \begin{bmatrix} \exp(-\lambda \mu) & 1 - \exp(-\lambda \mu) \\ u_i(\tau) \exp(-\lambda \mu) & 1 - u_i(\tau) \exp(-\lambda \mu) \end{bmatrix} \]  \hspace{1cm} (15)

From (13) and (15), for any imbedded time \( \tau \), we can construct a chain

\[ H_i(\tau) = \begin{bmatrix} F-J_n \Pi_{k=\tau} G_i(k) \\ \tau - 1 \Pi_{k=1} G_i(k) \end{bmatrix} \]  \hspace{1cm} (16)

where \( \Pi_{k=a} G_i(k) = \{ G_i(a), G_i(a+1), \ldots, G_i(b) \} \) and \( b = a \).
The chain in (16) specifies Data Station i’s buffer contents only at particular imbedded times \( \tau \) in every frame, and is, hence, homogeneous. Knowing \( H_{i}(\tau) \), we can determine the steady-state distribution of Station i’s buffer content \( \{ \pi_{im}(\tau), m = 0, 1, 2, \ldots, M, \tau = 1, 2, \ldots, F-J_{i} \} \). For any \( \tau \), we can get the probability that the buffer at Station i is nonempty to be \( q_{i}(\tau) \), while \( u_{i}(\tau) = 1/(1+((N_{d}-1)q_{i}(\tau))) \), and \( \pi_{i}(\tau) = 1/(1+\pi_{i}(\tau)) \). Using numerical methods, the above probabilities can be calculated. Letting the slot between \( \tau \) and \( \tau+1 \) be Data Slot \( \tau \), we can specify Station i’s throughput in that slot by

\[
s(i)(\tau) = u_{i}(\tau)q_{i}(\tau)
\]

Since all data stations in SUBS receive identical service, the overall data throughput for slot \( \tau \) is \( \sum_{i=1}^{F-J_{i}} n_{d}s(i)(\tau) \).

Removing the conditioning on \( n \) in (17), we find that the data throughput is

\[
S_{d} = \sum_{n=0}^{N_{v}} n \pi_{n}(\tau)
\]

where we have let \( n \), rather than \( J_{i} \), take on all possible values. Noting that the average number of slots carrying voice in a frame is

\[
E[V] = \sum_{n=0}^{N_{v}} n \pi_{n}(\tau)
\]

we can express the network’s total throughput (including both voice and data) by

\[
S_{v+d} = S_{d} + E[V]/\lambda.
\]

From (18) and the fact that \( \lambda \) is given, it follows that the blocking probability for data packets is \( P_{B} = (\lambda-S_{d})/\lambda \). This is also the blocking at each of the individual stations in SUBS.

For the delay analysis, we define \( T_{1}(\tau) \) to be the initial synchronization delay of a data packet arriving at Station i and synchronizing to Data Slot \( \tau \), and let \( T_{2}(\tau) \) be the additional number of slots required by the packet until its transmission is completed. We exclude the propagation delay from our analysis in order to keep our model simple and general. We note from [1] that the mean and variance of \( T_{1}(\tau) \) are

\[
E[T_{1}(\tau)] = \frac{\lambda_{e_{\tau}} - 1 + \exp(-\lambda_{e_{\tau}})}{\lambda_{e_{\tau}}(1-\exp(-\lambda_{e_{\tau}}))}
\]

and

\[
\text{Var}[T_{1}(\tau)] = \frac{(1-\exp(-\lambda_{e_{\tau}}))^{2} - (\lambda_{e_{\tau}})^{2}\exp(-\lambda_{e_{\tau}})}{\lambda_{e_{\tau}}^{2}(1-\exp(-\lambda_{e_{\tau}}))}
\]

respectively, where \( e_{\tau} \) is given by (14). We recall that (19) and (20) are exact for \( M=1 \), but approximate for \( M>1 \).

For \( M=1 \), it is a direct matter to verify that the distribution of \( T_{2}(\tau) \) for any \( J_{i} \) is given by

\[
P[T_{2}(\tau) = k|M=1] = \begin{cases} \Pi_{j=1}^{k+\tau-2} u_{j}(\tau) & \text{if } 1 \leq k \leq F-J_{i}-\tau+1 \\ \Pi_{j=1}^{F-J_{i}(\tau)} u_{j}(\tau) \end{cases}
\]

Removing the conditioning on \( n \) in (21), we find that the packet in question is transmitted in the frame of its arrival, while the other condition implies that the packet is transmitted \( w \) frames later. From (21), the moments of \( T_{1}(\tau) \) conditioned on \( M=1 \) follow [26].

For \( M>1 \), we first determine \( E[T_{2}(\tau)|M=1, m\leq M] \), where \( m \) is the buffer content when an arrival occurs at Station i. Letting \( \phi_{i}(\tau) = E[T_{2}(\tau)|M=1] \), we find that the oldest of these \( m \) packets takes \( \phi_{i}(\tau) \) slots to clear where \( \tau = \tau_{1} \). The next packet takes \( \phi_{i}(\tau) \) slots where \( \tau_{2} = (\tau_{1} + \phi_{i}(\tau)) \mod (F-J_{i}) \) since this packet starts contending for empty slots at the \( \tau_{2} \)th data slot in a frame. In a similar fashion, the average number of slots required by the rest of the \( m \) packets and, then, by the packet in question can be determined. For \( m=M \), we note that the arrival is accepted only if the oldest packet is transmitted in slot \( \tau \). Thus, we get

\[
E[T_{2}(\tau)|M>1, m=1] = \sum_{n=1}^{M-1} \phi_{i}(\tau_{w})
\]

where \( \tau_{w} = \begin{cases} \tau \text{ if } m=1 \\ \text{w} \end{cases} \). Removing the conditioning on \( m \) in (22), we get

\[
E[T_{2}(\tau)|M>1] = \sum_{m=1}^{M-1} \pi_{m}(\tau) E[T_{2}(\tau)|M>1, m=M]
\]

Since an arbitrary arrival is equally likely to occur anywhere in a frame, we easily remove the conditioning on \( \tau \) in (23). And, then, removing the conditioning on \( n \) also, we get

8B.3.5.

0817
We choose $D_{\text{max}} = 100$ ms and $R_v = 64$ kbps. From (2), we get $B = \leq 3200$ bits. Therefore, we set $B = 1000$ bits. Then, from (1), we find that $f = 15.62$ ms. We consider the worst-case layout first, and discuss the performance under the best-case layout later.

By using the estimator in (5), we show the percentage of speech loss suffered by the most downstream voice station, viz., Voice Station $N_v$ as a function of $N_v$ and $\delta$ for the worst-case layout in Fig. 5(a). The corresponding fraction of allocated bandwidth wasted due to the estimator is shown in Fig. 5(b). We observe that, as $N_v$ is increased (with $\delta$ fixed), the clipping increases but the wastage of slots decreases. This happens because, with increasing $N_v$, the diagonal terms of the matrix in (3) decrease while the off diagonal terms increase resulting in a loss of accuracy in the estimation process. We found that the average clipping experienced by the other voice stations was significantly lower than that for Station $N_v$. In particular, the clipping at Stations $N_v+1$ and $N_v+2$ were about two-thirds and one-half of that at Station $N_v$, respectively, while more than half of the most upstream voice stations suffered negligible clipping. We note that the plots in Figs. 5(a) and (b) are not exactly monotonic as shown. Instead, they have sawtooth-type profiles with discontinuous derivative at every $N_v$ but we omit their detail since our interest is only on the general behavior of clipping and $W_b$. For small values of $N_v$, when the voice subframe size is also $N_v$, we find that $W_b = 0.6$ because the speech activity factor is 0.4. We also found that the clipping decreases and $W_b$ increases with increasing $\delta$ as expected. In Table 1, we show the maximum $N_v$ that can be supported for given values of $\delta$ and the maximum tolerable percent clipping.

Using Figs. 5(a) and (b), we also verify some of the results of our analysis with those obtained by simulation for $\delta = 1$ and 2. We find that the clipping obtained by simulation is slightly higher than that predicted by analysis. This is because our analysis assumed that there is no clipping in frame $r-1$ as discussed in Section III(C). The $W_b$ obtained by both models, however, agree closely.

Next, we compare the performance under the estimators given by (5) and (12). The clipping under (12) was found to be much larger than that under (5). In particular, for $N_v=20$ and $\delta=1$, we found from our analysis that the clipping at Station $N_v$ was 2.7% under (12), while that under (5) was 0.3%. With simulation, however, this number was about 30% under (12), but, as discussed in Section III(C), such a high inaccuracy is not unexpected since our analysis, which assumes no clipping in frame $r-1$, becomes more inaccurate for (12). We also found that $W_b$ under (5) was larger than that under (12) by an amount $1.25/N_v$. This increase, for reasonable values of $N_v$, is negligible causing data performance under both disciplines to be almost identical. We remark that (12) would become accurate and attractive if $K_r-1$ is replaced by the number of voice stations ready to transmit packets in frame $r-1$. The latter quantity can be obtained by dedicating a portion of the channel bandwidth for out-of-band signaling by stations with voice packets. However, we choose (5) for estimating $L_f$ not only because it is simple, but also because it performs reasonably well.

To study data performance, we consider the estimator in (5), and initially choose a network with $N_v = 20$, $N_d = 10$, $R_w = 2$ Mbps, and $\delta = 1$. We show the average data packet delay, the delay variance, and the probability of blocking against the total network throughput for different values of $M$ in Figs. 6(a), 6(b), and 7, respectively. We find that the 20 voice stations consume about 25% of the channel capacity. For very light loads, the average data packet delay is about 2.8 slots. For this configuration, we find that the maximum channel throughput that can be achieved is about 97% of the channel data rate. For faster channels, the utilization can be increased even further towards unity. We find that the average delay and the variance increase, while the blocking decreases with increasing $M$ as in the pure data case (see [1]). In Figs. 6(a) and 7, we also show the average delay and blocking obtained for data packets by simulation of our protocol. For the simulation model, we use the $p_b$ obtained from analysis. We find a close match between our analytical and simulation results.

Via Figs. 8 through 11, we show the delay-throughput behavior for data as a function of system parameters. In Fig. 8, we let $N_d = 50$, $M = 1$, $R_w = 10$ Mbps, $\delta = 1$, and vary $N_v$. For very small $N_v$, we find that the channel becomes almost a pure data channel with $S_{v+d} \approx S_d$, and $E[D]$ at light load = 1.5 slots. As $N_v$ is increased, the fraction of the channel bandwidth used by voice increases and so does the average data delay. Since $F = 156$ for $R_w = 10$ Mbps, we find that, for $N_v = 156$, voice traffic takes up about 40% of the channel bandwidth.

We let $N_d$ take on different values in Fig. 9, but choose $N_v = 20$ and 156 while keeping $R_w$, $M$, and $\delta$ at their earlier values. For $N_v = 20$, as $N_d$ is increased, we find that the average delay increases since the buffering capability of the entire network also increases. Consequently, the blocking probability, which we do not show, decreases. For $N_v = 156$, we do, however, observe an anomalous result in Fig. 9, viz., $E[D]$ for $S_{v+d}$ is $84$ (0.4, 0.85) decreases with increasing $N_d$. This is because, with larger $N_v$, $L_f$ becomes longer. Thus, data packets that arrive during the voice subframe encounter longer delay. Such packets, for smaller $N_d$, face more blocking, and the unblocked packets are delayed longer on the average. When the data load is sufficiently high such that $S_{v+d}$ >
0.85, this anomaly disappears, and the normal effect, as explained earlier for \( N_V = 20 \), becomes dominant.

The effect of changing \( \delta \) is shown in Fig. 10 when \( R_W = 10 \text{ Mbps}, N_d = 50, M = 1, \) and \( N_V = 20 \) and 156. As \( \delta \) is increased, we find that the average delay increases slightly, while the maximum throughput drops a little since the voice subframe size is increased. At the same time, the data subframe size is reduced marginally.

Figure 11 is used to illustrate the effect of changing the channel bandwidth. We choose \( N_V = 20, N_d = 10, M = 1, \) and \( \delta = 1 \), and find that as \( R_W \) is increased, the fraction of the channel bandwidth used by voice drops since the number of voice calls is fixed. With increasing \( R_W \), we also find that the data delay decreases since there are more data slots now, and the maximum throughput moves closer to unity.

Finally, in Fig. 12, we show the difference in performance achieved under the best-case and the worst-case layouts. For the same values of \( N_V, N_d, M, R_W, \) and \( \delta \), we observe that the average delay is slightly lower for the best-case layout. For this layout, the channel throughput at heavy loads approaches unity, while that for the worst-case layout is a little less than unity because of the wastage of slots due to the estimator.

V. CONCLUSION

We have proposed and studied a new voice-data integration protocol for unidirectional bus networks. To avoid the capacity loss in round-robin-type access schemes, we used a framed architecture in which the voice subframe size was based on an estimator function, while the data subframe was accessed according to the \( \pi \)-persistent protocol. By suitable choice of the estimator, we found that the fraction of speech loss suffered by all voice stations can be maintained under a specified maximum. We also determined the fraction of the allocated bandwidth lost due to the estimator and found that the channel could be operated at a throughput close to unity. Further, we validated the results of our performance analysis via simulation.

For the analysis of voice, we set up an imbedded Markov chain for the number of ready voice stations at the frame boundaries and converted them to homogeneous chains which we solved to obtain the optimum \( \pi \) for each data slot. We observed that, by increasing \( M \), the blocking is reduced, while the average delay is increased. We also studied the effect of various system parameters on the delay-throughput performance of data traffic.

We note that the system performance is independent of the bus length and data rate (available under current technology). This property does not exist in the round-robin-type protocols for voice-data integration on unidirectional bus networks [13-17] in which the overhead per round increases with increasing bus length, or data rate, or both. Since our protocol is free from such overhead, it is very suitable for long unidirectional buses with high bandwidth applications.

REFERENCES


Fig. 1. Different voice and data station configurations in a SUBS network.

Fig. 2. Three-state speech source model [23].

Fig. 3. Activity on inbound channel showing voice and data subframes.

8B.3.8.
Fig. 4. Embedded times and associated transitions for Data Station i.

Fig. 5(a). Voice Performance: Average percent clipping.

Fig. 5(b). Voice Performance: Wasted allocated bandwidth.

Fig. 6(a). Data Performance: \( N_v = 20, N_d = 10, R_v = 64\, \text{kbps}, R_w = 2\, \text{Mbps}, \delta = 1 \).

Fig. 6(b). Data Performance: \( N_v = 20, N_d = 10, R_v = 64\, \text{kbps}, R_w = 2\, \text{Mbps}, \delta = 1 \).

Fig. 7. Data Performance: \( N_v = 20, N_d = 10, R_v = 64\, \text{kbps}, R_w = 2\, \text{Mbps}, \delta = 1 \).
Fig. 8. Data Performance: $N_v = 50$, $M = 1$, $R_v = 64$ kbps, $R_w = 10$ Mbps, $\delta = 1$.

Fig. 9. Data Performance: $N_v = 20$ and $156$, $M = 1$, $R_v = 64$ kbps, $R_w = 10$ Mbps, $\delta = 1$.

Fig. 12. Data Performance: Best-case vs. worst-case layout with $N_v = 20$, $N_d = 50$, $R_v = 64$ kbps, $R_w = 2$ Mbps, and $\delta = 1$.

Table 1. Maximum $N_v$ (by analysis) for different values of $\delta$ and maximum tolerable percent clipping.

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>52</td>
<td>87</td>
</tr>
<tr>
<td>2</td>
<td>110</td>
<td>170</td>
<td>220</td>
</tr>
<tr>
<td>3</td>
<td>250</td>
<td>290</td>
<td>470</td>
</tr>
</tbody>
</table>