An Integrated Services Token-Controlled Ring Network

PO-CHOI WONG, MEMBER, IEEE, AND TAK-SHING PETER YUM, SENIOR MEMBER, IEEE

Abstract—**A new integrated services token-controlled ring is proposed. This ring has the following distinct features. 1) It allows synchronous traffic such as voice and video to have a definite access to the channel within each packetization period (or frame). Hence, the transceiving unit needs only one transmit and one receive buffer for each synchronous stream. 2) It allows data messages to have a higher channel access priority provided that the synchronous traffic is not delayed by more than one frame. Urgent messages can therefore be transmitted with a minimum delay. 3) It supports variable rate data circuits. Simulation results show that the data message delay is much smaller than other integrated services schemes and the voice packet loss probability is essentially zero over a wide range of data and voice traffic loads.**

I. Introduction

A LOCAL area network (LAN) is a computer network that has a well-defined topology and access control connecting computers and communication equipment within a localized area, say an office, a building, or a campus. Compared to a digital PBX, a LAN has the advantages of 1) very low delay for data transmission, 2) high data transfer rate, 3) easy and flexible connections, 4) distributed control and therefore reliable and 5) low-cost. Hence, it is well-suited for bursty data traffic.

One exciting feature of LAN's is that they can be used to transmit not only computer data but also voice, facsimile, graphics, video, and other visual information. These LAN's are referred as integrated local area networks (ILAN) [1]. Together with the development of ISDN [2], [3], ILAN has become one of the hottest research and development topics in computer communications in recent years.

Different types of traffic have different service requirements. In terms of traffic to the ILAN's, they can be classified into

1) Short Asynchronous Data Traffic: Examples are interactive terminal enquiries, remote control messages, and sensing messages. Some of these may be urgent while others can tolerate a longer delay.

2) Bulky Asynchronous Data Traffic: These are the machine-to-machine traffic with large volume of data transfer such as file transfer, database updating, and high-volume printing. A substantially longer delay can be tolerated.

Manuscript received April 10, 1988; revised February 21, 1989.

P.-C. Wong is with the Department of Computing Studies, Hong Kong Polytechnic, Hong Kong.

T.-S. P. Yum is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.

IEEE Log Number 8927593.

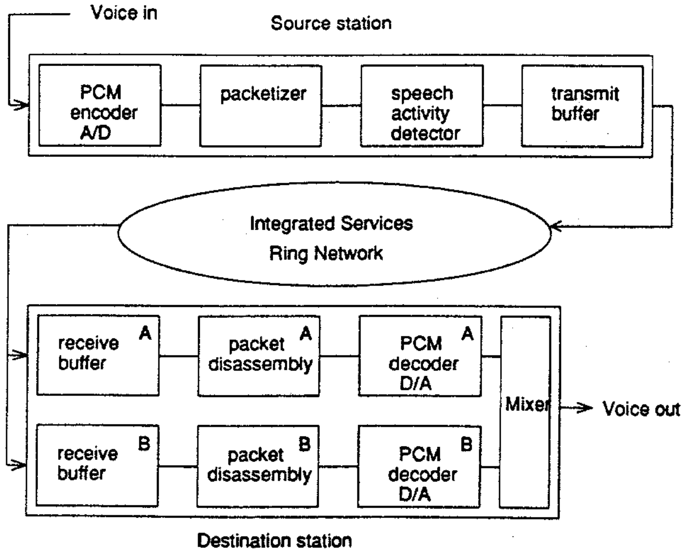

Fig. 1. Voice communication process (with conferencing).

3) Synchronous Stream Traffic: These are generated in real-time person-to-person calls and hence cannot tolerate a large transmission delay. For voice (video is similar except at a higher bandwidth), a one-way delay of more than 600 ms will be very annoying [4]. It can, however, tolerate a certain amount of degradation such as noise, clipping, compression, and occasional blocking without becoming objectionable [5].

Fig. 1 shows the voice communication process on an integrated services local area ring network. The voice waveform is sampled at 8 kHz and encoded into the 64 kbit/s PCM signal. In some ILAN protocols, the voice samples are transmitted directly using an assigned TDM channel (one byte per 125 μs frame) on the network. In most other ILAN protocols, the voice is transmitted in the form of packets where each packet consists of a number of voice samples within a packetization interval. The voice packet generation process may be synchronized to an external timing. The voice packet is then examined by a speech activity detector. Silent packets are discarded. Nonsilent packets are stored in a buffer and await transmission. At the destination, the packets are disassembled into individual samples to reconstruct the speech. The total delay of a voice packet is equal to the sum of the packetization, the queueing, and the transmission delays. A unique requirement on packetized voice is that the size of the voice packets must be kept small in order to reduce

0733-8716/89/0600-0670$01.00 © 1989 IEEE

the packetization delay. In Fig. 1, the receiver has two sets of voice packet decoders for the supporting of telephone conferencing. Their use will be discussed later.

To support international calls in ILAN's, the delay within an ILAN should be kept below some tens of milliseconds [6]. In some ILAN's, late packets (say, delayed by more than 50 ms [7]) are discarded. We can design an ILAN that supports two grades of voice service. For local and continental calls, the delay may be larger (say, 100 ms [8]). For international calls, the delay within an ILAN has to be substantially smaller. A detailed study on the delay issues in voice/data networks can be found in [9]. Conversation is inherently robust and can be reconstructed at the receiving end with acceptable quality provided that the voice packet loss is not serious. It is generally accepted that voice packets loss of less than 1 or 2 percent is tolerable [10]. However, if the voice is compressed before being packetized (such as the 32 Mbit/s ADPCM [11], [12]), loss of voice packets is objectionable. The effects of voice packet loss was studied in detail in [13].

Studies of LAN's that support integrated services may be classified into the following three types:

1) Synchronous Uniframe Approach [14]–[17]: Here, time is divided into PCM-like (125 μs) frames. The voice samples are transmitted one per frame using an assigned TDM channel.

2) Synchronous Slot Approach [1], [18]–[21]: Here, time is divided into fixed-size synchronous slots, each can accommodate a voice or a data packet.

3) Asynchronous Packet Approach [6], [7], [22]–[31]: This approach is normally used in lower speed ILAN's (1–10 Mbits/s) where the synchronous approach is less efficient in the integration process. All types of traffic are transmitted as packets and there is no explicit synchronization among the stations.

To ensure a continual regeneration of speech, conventional protocols using the asynchronous packet approach have introduced one or more of the following limitations.

1) Higher priority is given to the voice packets [6], [28], [31]. This makes the data delay heavily dependent on the amount of voice traffic on the ring.

2) Data messages have to be broken into small voice-packet-like units, each with a header and check bits attached [6], [25], [30], [31].

3) The voice and data packets are transmitted in alternate cycles [25], [28], [30]. Urgent data messages have to wait for the data cycle before they can be transmitted. Synchronization of cycles to the voice packet generation process is also difficult.

4) Buffers are needed for the incoming voice packets for smoothing out the random voice packet delays [6], [28].

5) Time-stamping on voice packets is needed since the delays on voice packets are nondeterministic [6], [7], [25], [28], [30], [32], [33].

6) Discard of voice packets may be necessary if their delays exceed a certain limit [6], [7], [28], [32].

We propose in the following a protocol for the ring network that attempts to avoid these limitations. The protocol has the following new features.

1) Priority is given to data messages to minimize their delay as long as the voice packets are not delayed by more than one packetization period.

2) All voice packets are delivered within a packetization period. Within that time period, earlier or later delivery is, of course, immaterial.

3) The loss of voice packets can be completely avoided.

4) Variable size data messages and variable rate data circuits are supported.

We will first discuss how the protocol can support integrated voice/data services and then provide simulation results to substantiate the many nice properties of this protocol.

II. Protocol Description

A. Ring Configuration

The ring as shown in Fig. 2 requires one of the stations to act as the *monitor* which handles the ring management functions such as ring initialization, ring reset, and initialization of the tokens. This is normally required in a token ring network. All stations have a 1 bit station latency except the *monitor* which has a 30 bit latency. The *monitor* can also take the responsibility of a gateway to support switching to and from the outside sources. Another station may take the role of a *backup monitor* by a small duplication of hardware at the ring interface (transparent to the station). With a *monitor* to handle all the call management functions such as call-setup, call-release, and recording of statistics, the processing in all the other stations can be much simplified.

B. Timing Consideration

Time is divided into frames. The length of a frame is equal to the packetization period of the speech signal. Therefore, each voice station will generate one voice packet per frame during the talkspurt period. Each frame consists of K slots and each slot can accommodate one voice packet plus the overheads of interstation propagation delay and token transmission time.

As an example of how K is determined, we consider a 400 station ring on a 2 km cable operating at 10 Mbits/s. Let the packetization period be 20 ms. With 64 kbit/s PCM encoding and a packet header of 168 bits added [34], the size of a voice packet is 1448 bits. Let the token size be 30 bits. Then, the total ring latency is the sum of the roundtrip propagation delay, the *monitor's* latency, and the 1 bit delay for each of the 400 stations, which is 100 bits + 30 bits + 400 bits = 530 bits. Also, each slot in the frame must contain 1448 bits of voice packet and 30 bits of token transmission delay. Therefore, the total number of bits in a 20 ms frame is equal to (K)(slot size in bits) + (ring latency in bits), or $2 \times 10^5 = K(1448 + 30) + 530$. K is therefore 134 slots.

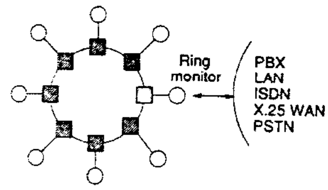

Fig. 2. Ring topology (with a ring monitor).

SD	DS	Control				ED
8-bits	10-bits	R	D	F	T	8-bits

(a)

Legends:

a. SD, ED : start/end delimiter
b. DS : data slot count
c. R : reservation ($R=1$ - normal, $R=0$ - reserved)
D : token double visit ($D=1$ - second, $D=0$ - first)
e. F : token status ($F=1$ - *free*, $F=0$ - *busy*)
T : type ($T=1$ - a voice packet, $T=0$ - a data packet)

Message Header			Message Body				
SD	DS	Control	DST	SRC	INFO	CRC	ED

(b)

Fig. 3. (a) Token formats. (b) Format of voice packets and data messages.

C. Token and Message Formats

The token and message formats are shown in Fig. 3. A 10 bit DS field (data slot count) in the token is used to track the number of available slots for data in each frame and is initialized at the beginning of each frame to DS_o. This ensures that the voice packets are not delayed by more than DS_o data slots. In the above example, if 80 slots are assigned for voice packets, DS_o will be $K - 80$ or 54 slots. The *monitor* keeps a 10 bit counter register called *CLOCK*. *CLOCK* is decremented by one every time slot and gets reset to K at the beginning of each frame.

The token and the message header contain four control bits denoted as R, F, T, and D. R is a priority bit to be set by the data stations for reserving the next transmission right. $F = 0$ or 1 indicates whether the token is *free* or *busy*. T is 0 for data messages and 1 for voice packets. The D-bit is set to 1 by the *monitor* whenever a free token with $D = 0$ passes by and set to 0 by the station releasing the token after a transmission. Whenever a free token with $D = 1$ comes back to the *monitor*, the *monitor* can conclude that the free token is visiting it the second time without being seized by any station; and hence all voice stations must have transmitted their current voice packets. The functions of the DS, *CLOCK*, and the control bits will be discussed in more detail in the next section. Call management functions such as call setup and release are handled by control packets. Control packets usually have a higher priority than the other traffic.

D. Station Access Scheme

At the start of each frame, the *monitor* initializes the token with $DS = DS_o$, $D = 0$, $R = 1$, and $F = 1$ before releasing it to the ring.

1) Data Message Access Procedures: When a station has a data message of length between $m - 1$ and m slots to transmit, it waits for the arrival of a *free* token ($F = 1$) and checks the value of DS in the token. If $DS \geq m$ or the number of data slots is sufficient, it will 1) seize the token by setting the F-bit to 0, 2) transmit the message with $T = 0$, and 3) release a *free* token with DS decreased by m. On the other hand, if $m < DS$, the station just forwards the token to the downstream station.

If a station has an urgent data message to send, it can reserve the ring for a priority transmission. When a voice packet header passes by and the DS is large enough, the reserving station will set the R-bit to 0. When the header gets back to the transmitting station, it will release a token with $R = 0$ to signify the downstream stations that the ring has been reserved. The data station making the reservation can now seize the token and initiate its transmission. Afterwards, it will release a *free* token ($F = 1$, $R = 1$). If more than one data station wants to make reservation, they will reserve and transmit one by one according to their order on the ring.

2) Voice Call Setup Procedures: The *monitor* keeps a record of ongoing calls in the ring. When a station wants to initiate a new call, the station sends a *call-request* packet to the *monitor*. The *monitor* accepts the call or not by looking up the table to see if 1) the called station is free, and 2) the call limit is not reached. If the call is not accepted, then either a *circuit-busy* packet or a *station-busy* packet will be sent to the calling station. If both 1) and 2) are satisfied, the *monitor* will send out a *ringing* packet to be read by the calling and the called stations. The called station will start "ringing." When the call is answered, an *accept* packet is transmitted to the calling station and the *monitor*. The *monitor* updates its table and the conversation can begin. When either party hangs-up the phone, a *call-release* packet is transmitted to the other party and to the *monitor* (for updating table).

3) Voice Conference Call Setup Procedures: With the broadcasting nature of the ring, conference calls can be implemented with slight modifications to the ordinary call procedures. If a station wants to set up a conference call with a set of other stations (the called stations), it sends a *conference-request* packet to the *monitor*. If the call is accepted, the *monitor* will send a *ringing* packet (with multiple addresses) to the called stations as well as to the calling station. When one of the called users picks up the phone, an *accept* packet will be transmitted to the calling station and the *monitor*. The conference call may start. Other called stations can join the conversation by transmitting an *accept* packet to the calling station and the *monitor*. The *monitor* updates the table as the conference call proceeds.

During a conference call (including two-party call), other users can be invited to join the conversation. The "inviting" station just transmits a *ringing* packet to the new station. When the call is picked up, an *accept* packet is transmitted to the *monitor* and the "inviting" station.

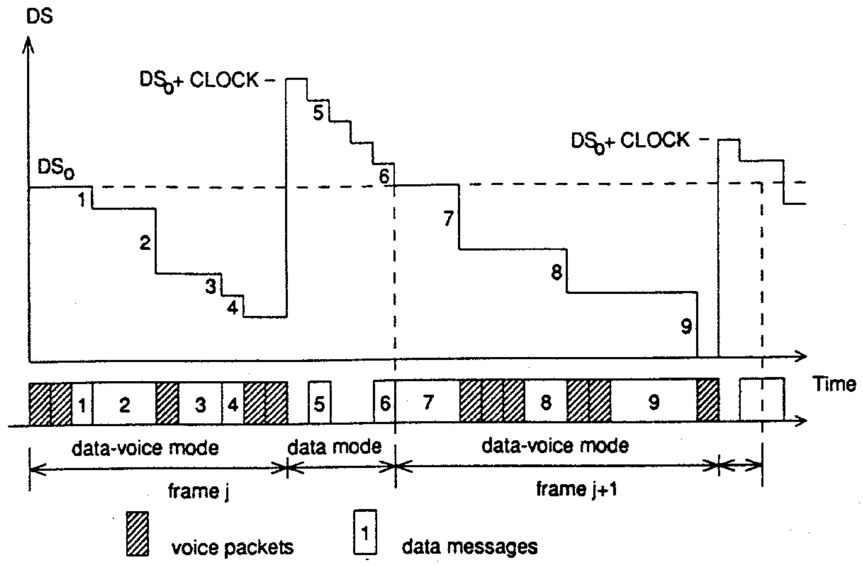

Fig. 4. The two ring operating modes.

Note that the channel bandwidth requirement of an n-party conference call is only slightly greater than a two-party call. This is because during a conference call, the probability that three or more users talking simultaneously is very small.

4) Voice Packet Transmission Procedures: Once a call is set up, the active voice station may either be in the *talking* mode generating one voice packet per frame or in the *silence* mode where no packet is generated. In our protocol, all voice packets are generated at the beginning of a frame (the frame boundary can be derived from the token and will be explained later). The voice packets generated from the *active* voice stations are transmitted one by one following the station order in the ring. Each station will wait for a free token ($F = 1$) to arrive, seize it by setting the F-bit to 0, and transmit the voice packet with $T = 1$. After that, the station releases a free token with $F = 1$ and $D = 0$.

E. Ring Operating Modes

At the beginning of each frame, the ring is operated in the *data-voice* mode where both voice and data are accepted for transmission (Fig. 4). Whenever a data message of varying size is transmitted, DS is decremented proportionally. Reservation by data stations is allowed as long as $DS \geq m$. In time, when all the voice packets are transmitted, the remaining slots in the frame can be used exclusively for data transmission. We said the ring has entered the *data* mode.

The *data* mode is initiated by the *monitor* when a free token visited the *monitor* twice (indicated by $D = 1$) before the frame ends. It does so by setting $DS = CLOCK + DS_o$ (or the sum of the remaining slots in this frame and the number of available data slots in the next frame). In *data* mode, $CLOCK$ is also decremented every timeslot and whenever a token comes back to the *monitor*, DS is again set to $CLOCK + DS_o$. DS therefore carries the current number of data slots available. Also, in *data* mode, data transmission may cross a frame boundary. This only results in the delay of the *data-voice* mode and no updating of $CLOCK$ and DS is needed.

When $CLOCK$ reaches zero, the frame ends. The *monitor* immediately resets $CLOCK$ to K to start a new frame. The ring will change back to *data-voice* mode.

F. Speech Regeneration Process

At the receiving side, each arriving voice packet is first stored in a buffer and then transferred to the vocoder to regenerate the speech waveform. Since all voice packets are generated at the beginning of a frame and delivered before the end of the frame, speech continuity is ensured by delaying the use of the first talkspurt packet until the end of a frame. No time-stamping is necessary since all nonsilent packets will arrive within the frame and will be used at the end of the frame. The voice packet reception and speech regeneration process is shown in Fig. 5.

In a conference call, two talkspurts (analog waveforms) from different speakers can be received simultaneously. At the receiving end (Fig. 1) the two packet streams are converted into analog waveforms and mixed to be output to the speaker so that the user can hear the interrupted speech. Usually when a speech is interrupted, the more dominant speaker will continue while the others will yield. Only two voice decoders are necessary in the receiver because a jam of more than two voices can hardly be audible. A station will refrain from transmitting its own voice packets when two packets are being decoded.

G. Synchronization

The protocol requires all voice stations to synchronize their packet generation to the beginning of each frame. A voice station can derive the timing information by monitoring the DS value on tokens or message headers that pass by. In *data-voice* mode, the DS value is always

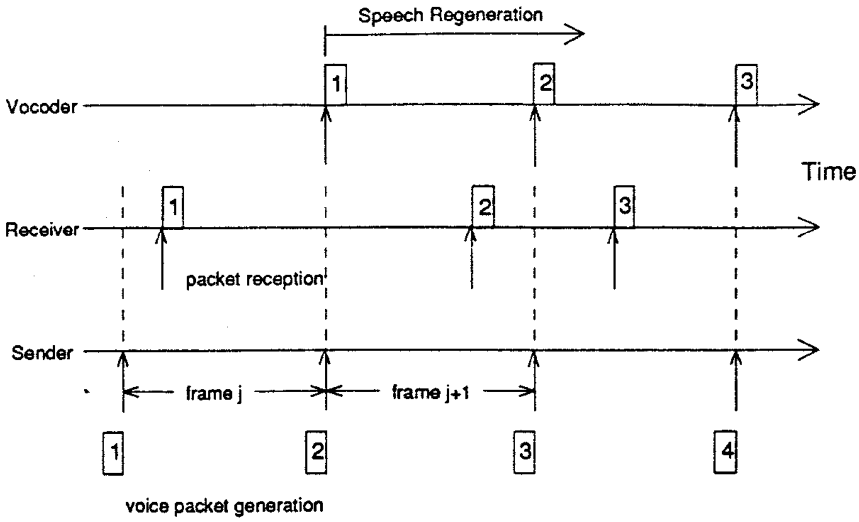

Fig. 5. The transmission and reception of voice packets.

smaller or equal to DS_o while in *data* mode, the DS is equal tc $CLOCK + DS_o$. So by subtracting DS_o from the DS value, a voice station can know the current $CLOCK$ value and deduce the starting time of the next frame.

H. The Choice of DS_o

In order that no voice packet will have to wait for more than one packetization period, DS_o in frame i should be equal to $K - n(i)$, where K is the total number of time slots in a packetization period and $n(i)$ is the total number of voice packets generated in frame i. Since DS_o must be set at the beginning of the frame, $n(i)$ must be estimated. We propose two estimators: one is *dynamic* and one is *fixed*. In the fixed approach, the *monitor* uses the number of active calls to estimate $n(i)$ according to the talkspurt-silence statistics. In the dynamic approach, the *monitor* counts the number of voice packets $n(i - 1)$ transmitted in the previous frame, adds Δ slots to it (to allow for statistical fluctuations), and uses it as an estimator for $n(i)$. The value of DS_o at frame i is $K - n(i - 1) - \Delta$. The fixed approach is simpler and does not require the *monitor* to count voice packets. But it causes a larger data delay than the dynamic approach [37]. Both estimators will be derived in Section III.

I. Variable Rate Data Circuits

Circuits of various data rates can be set up by requesting the *monitor* to assign one or more time slots to the station every frame. These data slots are given a higher priority than voice and so are virtual TDM slots. As an example if a 144 kbit/s (ISDN $2B + D$) circuit is required, three slots per frame can be assigned. Even if the data cannot occupy the entire three slots, the remaining slot time is not wasted because it can be saved for other transmissions in the *data* mode. If a lower speed data circuit, say 16 kbits/s is required, a station can request one slot every four frames.

J. Summary

The ring is either operating in the *data-voice* mode or the *data* mode. Voice packets can always be transmitted within each frame so there will be no loss of voice packets if the number of voice slots allocated is sufficient. Data messages can be transmitted with a higher priority over voice packets so the data delays are reduced.

III. Allocation of Voice Slots

Let N_V be the number of *active* voice stations. If we allocate N_V slots for voice transmissions, there will be no voice packet loss. However, with the TASI advantage [35], the number of slots required could be much smaller. In this section, the dynamic and fixed estimators for the number of voice slots are derived.

A. Voice Packet Loss Probability P_L

We assume that the durations of talkspurts intervals and silence intervals are exponentially distributed with means $1/u = 0.17$ and $1/v = 0.41$ s, respectively [36]. Let $\tilde{n}$ be the number of stations in *talking* mode. Since each *talking* station generates one packet, the number of voice packets generated is also $\tilde{n}$. The evolution of $\tilde{n}$ from frame to frame can be described by a birth-death process (Fig. 6) with birth rate $\lambda(n)$ and death rate $\mu(n)$ being

$$\lambda(n) = (N_V - n)v$$
$$\mu(n) = nu; \qquad 0 \le n \le N_V. \tag{1}$$

The steady-state probabilities $p_n = \Pr\{\tilde{n} = n\}$ can be solved as

$$p_n = p_{n-1}\frac{\lambda(n-1)}{\mu(n)} = p_0 \prod_{i=1}^{n} \frac{\lambda(i-1)}{\mu(i)}. \tag{2}$$

Using the normalization equation and letting $r = v/u$, p_0

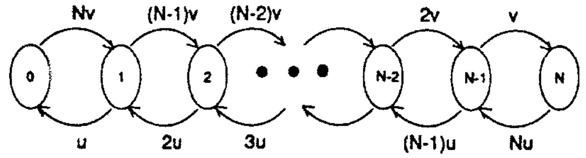

Fig. 6. The number of stations in talking mode modeled as a birth–death process.

is solved as

$$p_0 = \left(1 + \sum_{n=1}^{N_V} \prod_{i=1}^{n} \frac{\lambda(i-1)}{\mu(i)}\right)^{-1} = \left(1 + \sum_{n=1}^{N_V} \frac{N_V!}{(N_V - n)!\,n!} r^n\right)^{-1} = (1 + r)^{-N_V}. \quad (3)$$

Substitute into (2), we have

$$p_n = \frac{\binom{N_V}{n} r^n}{(1 + r)^{N_V}} \qquad 0 \le n \le N_V. \quad (4)$$

Let k slots be allocated for voice transmission. Then the voice packet loss probability P_L is

$$P_L = \frac{\text{Expected number of lost packets in one packetization period}}{\text{Expected number of packets generated in one packetization period}} = \frac{\sum_{n=k+1}^{N_V} (n-k)p_n}{\sum_{n=0}^{N_V} np_n} = \frac{v + u}{vN_V} \sum_{n=k+1}^{N_V} (n-k)p_n. \quad (5)$$

B. Fixed Estimator n_f

Here a fixed number of slots $n_f(\alpha)$ is allocated for voice transmission such that voice packet loss probability P_L is less than α. More precisely, $n_f(\alpha)$ is the minimum value of k in (5) such that $P_L < \alpha$. Table I shows n_f(0.1 percent) and n_f(0.01 percent) for N_V equals to 90, 180, 270, and 360, respectively.

C. Dynamic Estimator n_d

Let T be the frame size. For all active voice stations, let

$$q_t = \Pr\{\textit{talking} \text{ mode in the next frame} \mid \textit{silence} \text{ mode in the current frame}\} = 1 - e^{-vT}$$

$$q_s = \Pr\{\textit{silence} \text{ mode in the next frame} \mid \textit{talking} \text{ mode in the current frame}\} = 1 - e^{-uT}.$$

Then, given the number of *talking* stations $n(i-1)$ in frame $(i-1)$ is m, the total number of stations $\tilde{k}_t(i)$ entering the *talking* mode in frame i has distribution

$$\Pr\{\tilde{k}_t(i) = j \mid \tilde{n}(i-1) = m\} = \binom{N_V - m}{j} q_t^j (1 - q_t)^{N_V - m - j}. \quad (7a)$$

Similarly, the total number of stations $\tilde{k}_s(i)$ entering the *silence* mode in frame i has distribution

$$\Pr\{\tilde{k}_s(i) = j \mid \tilde{n}(i-1) = m\} = \binom{m}{j} q_s^j (1 - q_s)^{m-j}. \quad (7b)$$

The total number of voice packets generated in frame i is

$$\tilde{n}(i) = \tilde{n}(i-1) + \tilde{k}_t(i) - \tilde{k}_s(i). \quad (8)$$

Let the dynamic estimator be $\tilde{n}_d(\alpha) = \tilde{n}(i-1) + \Delta$. Then given $\tilde{n}(i-1) = m$, the number of voice packet loss $\tilde{l}$ in frame i is

$$\tilde{l}\,\big|_{\tilde{n}(i-1)=m} = [\tilde{n}(i) - \tilde{n}_d(\alpha)]^+ = [\tilde{k}_t(i) - \tilde{k}_s(i) - \Delta]^+. \quad (9)$$

Therefore,

$$P_L(\Delta) = \frac{\sum_{m=0}^{N_V} E[\tilde{l} \mid \tilde{n}(i-1) = m] p_m}{E[\tilde{n}]}. \quad (10)$$

Here again, Δ is chosen such that $P_L(\Delta) \le \alpha$.

Table I shows Δ for N_V = 90, 180, 270, and 360, respectively, at P_L = 0.1 percent and P_L = 0.01 percent. The average number of slots allocated for voice n_d ($E[\tilde{n}] + \Delta$) is also shown to be significantly smaller than the corresponding n_f's. Since the lengths of talkspurt and silence intervals are relatively long compared to the pack-

TABLE I
COMPARISONS OF FIXED AND DYNAMIC ESTIMATORS

	Fixed Estimator		Dynamic Estimator			
Number of active voice stations N_v	n_f (0.1%)	n_f (0.01%)	Δ (0.1%)	n_d (0.1%)	Δ (0.01%)	n_d (0.01%)
90	37	41	5	32	7	34
180	68	73	7	60	10	63
270	98	104	8	89	11	92
360	127	134	9	115	13	119

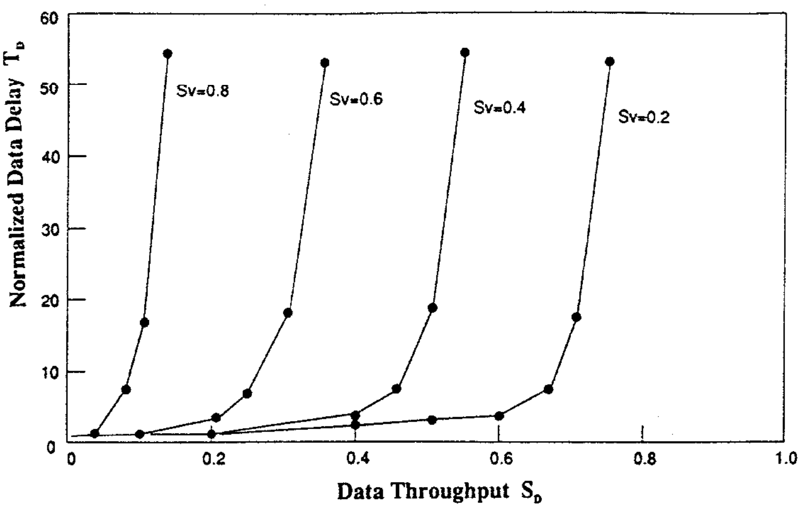

Fig. 7. Delay-throughput characteristics of data messages.

etization interval, the dynamic approach provides a more accurate estimate. The average data delay is also slightly better for the dynamic approach since a larger $DS_o = K - n_d$ can be assigned for data transmission. This is verified by simulation [37].

Note that in the derivation of the above estimators, the data traffic occupy all unassigned slots. In practice, all unoccupied data slots can be used for the voice packets. Extensive simulation shows that all the estimators in Table I give a loss-free operation as long as the system capacity is not exceeded. Worse comes to worse, if a station did lose a packet, it can prevent a consecutive loss by reserving a slot for its voice packet in the next frame.

IV. Simulation Results and Discussions

In order to derive the performance of the integrated services ring under various operating conditions, a continuous-time, event-driven simulation program is implemented in Pascal. The specifications of the simulation model are the following.

1) All stations are uniformly distributed on the ring, i.e., the station-to-station propagation delay is the same for all stations. We assume the number of active voice stations remains unchanged for the entire simulation run and the call destination is uniformly distributed on the ring. For data stations, the probability that a message in station i ($i = 1, 2, \cdots, N$) is destined for station j ($j \neq i$) is $1/(N-1)$ in an N-node ring.

2) The interarrival time and the size of data messages are exponentially distributed with mean $1/\lambda$ and $E[X]$, respectively. Message header size H is 168 bits. Let there be N_D data stations. The total data traffic load is given by $N_D\lambda(E[X] + H)$. The normalized data throughput S_D is therefore $N_D\lambda(E[X] + H)/C$, where C is the cable data rate.

3) The durations of talkspurts and silence intervals are exponentially distributed with means $1/u = 0.17$ and $1/v = 0.41$ s, respectively [36]. Other suggested parameters are 1.23 and 1.77 s in [4] and 0.185 and 1.31 s in [6]. The packetization interval T (frame size) is 20 ms. The PCM encoding rate is 64 kbits/s. Voice packet size V is 1448 bits (including a 168 bit header). Let there be N_V active voice stations and let $r = v/(u + v)$ be the talkspurt duty ratio. The normalized voice throughput S_V is given by $rVN_V/(TC)$.

4) Due to its better performance, the dynamic estimator is used for allocating voice slots.

A. Data Delay

Fig. 7 shows the average data delay T_D (normalized to average message transmission time) versus S_D in a 10 Mbit/s, 2 km ring with $S_V = 0.2$, 0.4, 0.6, and 0.8, respectively. The message size is exponentially distributed with mean 4 kbits/s. The total throughput $S_D + S_V$ can reach 0.94 when the ring is heavily loaded.

We observe that T_D is very small until the system capacity is reached. Take an example, when $S_V = 0.6$ and S_D is 0.2, the data delay is only 2.1 message transmission time, which is 0.84 ms. The small data delay is due to the higher priority of data messages in the ring.

B. Maximum Number of Active Voice Stations

If the ring is used to support voice traffic only, the maximum number of active voice stations N_V supported can be much larger than the number of slots K available on a ring. However, if N_V is too large, the probability of voice packet loss may be significant. We can determine the loss probability for a given N_V by substituting $k = K$ in (5). Fig. 8 shows the probability of packet loss P_L versus N_V in a 10 Mbit/s, 2 km ring. Simulation results are also shown for comparison. With $K = 134$, $N_V = 435$ for $P_L = 1$ percent and $N_V = 400$ for $P_L = 0.1$ percent. In a 5 Mbit/s ring with $K = 67$ slots, $N_V = 207$ for 1 percent loss and $N_V = 178$ for 0.1 percent loss. If we assume that the traffic load per subscriber is around 0.2 erlangs in the busy hour [27], a total population of around 2000 and 1000 subscribers can be supported for 10 and 5 Mbit/s rings, respectively. Even if 40 percent of the bandwidth is allocated for data, the 10 and 5 Mbit/s rings can still support 1200 and 600 subscribers, respectively. This

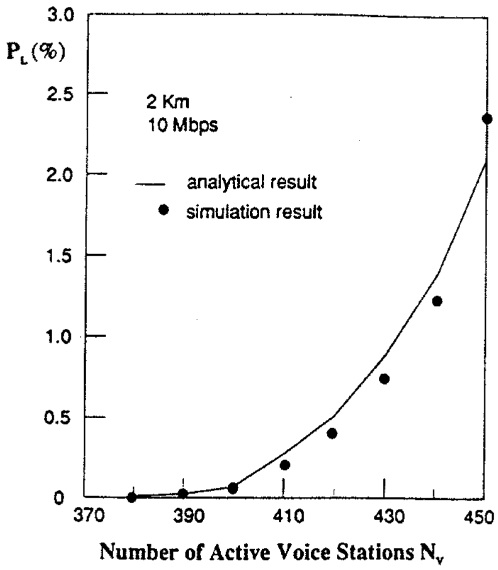

Fig. 8. Voice packet loss probability P_L (no data traffic).

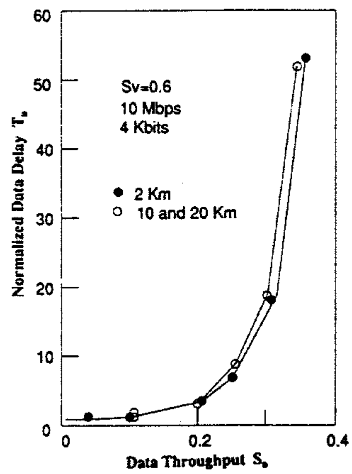

Fig. 9. Message delay comparisons: 2, 10, and 20 km cables.

makes the integrated services ring an attractive alternative to the digital PBX.

C. Cable Length

Fig. 9 compares T_D for rings with cable lengths equal to 2, 10, and 20 km. The data rate is 10 Mbits/s and the message size is 4 kbits. The delays of the 10 and 20 km rings are only slightly larger. Similar results are obtained for rings operating at 5 and 20 Mbits/s and message sizes of 1 and 16 kbits. The reason is that under heavy traffic condition, the token walk time is only from one station to its neighbor since all voice packets are generated at the beginning of a frame. This protocol therefore is also feasible on rings covering a larger geographic area.

D. Message Size

Fig. 10 shows that the T_D for rings with shorter messages is larger than that with longer messages. This is obviously due to the higher ring overhead for shorter messages. Compared to other protocols requiring the breaking

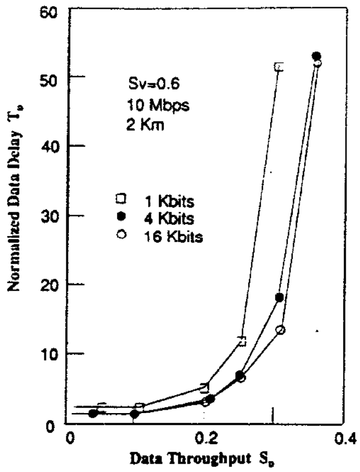

Fig. 10. Message delay comparisons: exponential message lengths of mean 1, 4, and 16 kbits.

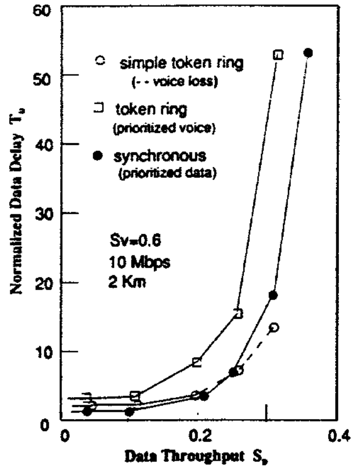

Fig. 11. Message delays of three ring protocols, $S_V = 0.6$.

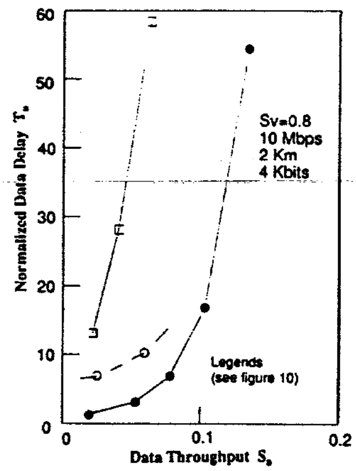

Fig. 12. Message delays of three ring protocols, $S_V = 0.8$.

up of messages into smaller units, our protocol has a definite advantage.

E. Comparison to Other Schemes

Figs. 11 and 12 compare the integrated services ring at $S_V = 0.6$ and 0.8 with 1) simple token ring and 2) prioritized-voice token ring. For the simple token ring, a significant voice packet loss (> 1 percent) is observed at $S_V = 0.6$ and $S_D > 0.2$. At $S_V = 0.8$, the situation is even worse with voice loss observable when $S_D > 0.05$. Without priority, data delay is also significantly larger. For the prioritized-voice token ring at $S_V = 0.6$, the data delay is significantly larger than our proposed integrated services ring. At $S_V = 0.8$, the data delay increases drastically.

V. Conclusion

In this paper, we proposed a ring protocol which allows voice and data traffic to coexist in harmony. Simulation results showed that the data message delay is much smaller than other integrated services schemes. Urgent messages can be transmitted with a higher priority over voice, which is usually not available in other integrated voice/data networks. Since the voice packet delay is bounded within one packetization period, no time-stamping is needed and the voice loss can be completely avoided by reserving a sufficient number of slots. Continual speech reception is possible by synchronizing the speech regeneration process to the end of each frame. Since the ring is synchronized, gateway switching to external circuit-switched and packet-switched networks is very simple.

Acknowledgment

We would like to thank the reviewers for their valuable comments. We also like to thank the Hong Kong Polytechnic for supporting the Ph.D. study of the first author. Thanks are also due to W. Y. Choy, K. H. Leung, and C. K. Poon for developing the simulation program, and to Mrs. P. C. Wong for preparing the figures and graphs.

References

[1] A. Patir, T. Takahashi, Y. Tamura, M. ElZarki, and A. A. Lazar, "An optical fiber-based ILAN for MAGNET's testbed environment," *IEEE J. Select. Areas Commun.*, vol. SAC-3, pp. 872-881, Nov. 1985.

[2] K. Habara, "ISDN: A look at the future through the past," *IEEE Communications*, vol. 26, pp. 25-32, Nov. 1988.

[3] J. Lane, *The Integrated Services Digital Network*, NCC Publications, 1987.

[4] P. T. Brady, "A technique for investigating on-off patterns of speech," *Bell Syst. Tech. J.*, vol. 44, Jan. 1965.

[5] T. M. Chen and D. G. Messerchmitt, "Integrated voice/data switching," *IEEE Communications*, vol. 26, pp. 16-26, June 1988.

[6] N. F. Maxemchuk, "A variation on CSMA/CD that yields movable TDM slots in integrated voice/data local networks," *Bell Syst. Tech. J.*, pp. 1527-1550, Sept. 1982.

[7] T. Suda and T. T. Bradley, "Packetized voice/data integrated transmission on a token-passing ring local area network," in *Proc. INFOCOM'87*, pp. 836-845.

[8] J. D. Detreville, "A simulation-based comparison of voice transmission on CSMA/CD networks and on token buses," *AT&T Bell Labs. Tech. J.*, pp. 763-774, June 1983.

[9] J. G. Gruber, "Delay related issues in integrated voice and data networks," *IEEE Trans. Commun.*, vol. COM-29, pp. 786-800, June 1981.

[10] S. R. Sachs, "Alternative local area network access protocols," *IEEE Communications*, vol. 26, pp. 25-45, Mar. 1988.

[11] N. Psimenatos, J. Gruger, G. Goddard, and D. Mondor, "Network considerations for the introduction of 32 Kbps ADPCM systems in telecommunications networks," in *Proc. IEEE GLOBECOM'84*, 1984.

[12] W. R. Daumer, X. Maitre, P. Mermelstein, and I. Tokizawa, "Overview of the ADPCM coding algorithm," in *Proc. IEEE GLOBECOM'84*, 1984.

[13] N. S. Jayant and S. W. Christensen, "Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure," *IEEE Trans. Commun.*, vol. COM-29, pp. 101-109, Feb. 1981.

[14] J. R. Brandsma, A. A. M. L. Bruekers, and J. L. W. Kessels, "PHILAN: A fiber-optic ring for voice and data," *IEEE Communications*, vol. 24, pp. 16-22, Dec. 1986.

[15] L. M. Casey, R. C. Dittburner, and N. D. Gamage, "FXNET, a backbone ring for voice and data," *IEEE Communications*, vol. 24, pp. 23-28, Dec. 1986.

[16] F. E. Ross, "FDDI-A tutorial," *IEEE Communications*, vol. 24, pp. 10-23, May 1986.

[17] A. Brosio, F. Gagliardi, L. Lambarelli, G. Panarotto, D. Roffinella, and M. Sposini, "A reconfigurable high-speed optical systems for integrated local communications," *IEEE J. Select. Areas Commun.*, vol. SAC-3, pp. 825-833, Nov. 1985.

[18] J. O. Limb and L. E. Flamm, "A distributed local area network packet protocol for combined voice and data transmission," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 926-934, Nov. 1983.

[19] R. M. Falconer and J. L. Adams, "Orwell: A protocol for an integrated services local area network," *British Telecom Technol.*, pp. 27-35, Oct. 1985.

[20] D. Roffinella, C. Trinchero, and G. Freschi, "Interworking solutions for a two-level integrated services local area network," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1444-1453, Dec. 1987.

[21] R. M. Newman, Z. L. Budrilkis, and J. L. Hullett, "The QPSX MAN," *IEEE Communications*, vol. 26, pp. 20-28, Apr. 1988.

[22] G. J. Nutt and D. L. Bayer, "Performance of CSMA/CD networks under combined voice and data loads," *IEEE Trans. Commun.*, vol. COM-30, pp. 1-11, Jan. 1982.

[23] J. M. Musser, T. T. Liu, L. Li, and G. J. Boggs, "A local area network as a telephone local subscriber loop," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1046-1053, Dec. 1983.

[24] D. H. Johnson and G. C. O'Leary, "A local access network for packetized digital voice communication," *IEEE Trans. Commun.*, vol. COM-29, May 1981.

[25] J. S. Meditch and Y. Zhao, "Framed TDMA/CSMA for integrated voice-data local area network," in *Proc. ICC'85*, pp. 10-17.

[26] P. Sen, "Modeling and performance analysis of a hybrid protocol for voice-data integration in local area networks," in *Proc. INFOCOM'87*, Apr. 87, pp. 981-986.

[27] J. M. Appleton and M. M. Peterson, "Traffic analysis of a token ring PBX," *IEEE Trans. Commun.*, vol. COM-30, pp. 417-422, May 1986.

[28] T. Saydam and A. S. Sethi, "Performance evaluation of voice-data token-ring LANs with random priorities," in *Proc. INFOCOM'85*, pp. 326-332.

[29] F. E. Ross, "FDDI-fiber, farther, faster," in *Proc. INFOCOM'86*, Miami, FL, Apr. 1986, pp. 323-330.

[30] M. Fine and F. A. Tobagi, "Packet voice on a local area network with round robin service," *IEEE Trans. Commun.*, vol. COM-30, pp. 906-915, Sept. 1986.

[31] W. Bux, F. H. Closs, K. Kuemmerle, H. J. Keller, and H. R. Mueller, "Architecture and design of a reliable token-ring network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 756-765, Nov. 1983.

[32] P. C. Wong and T. S. Yum, "An integrated services local area network," in *Proc. INFOCOM'87*, Apr. 1987, pp. 999-1008.

[33] D. Karvelas and A. Leon-Garcia, "A performance comparison of voice/data token ring protocols," in *Proc. INFOCOM'87*, pp. 846-855.

[34] Local Area Network Standards: Token-passing ring access methods and physical layer specifications, ANSI/IEEE 802.5, 1985.

[35] K. Bullington and J. M. Frazer, "Engineering aspects of TASI," *Bell Syst. Tech. J.*, vol. 38, pp. 353-364, 1959.

[36] J. G. Gruber, "A comparison of measured and calculated speech temporal parameters relevant to speech activity detection," *IEEE Trans. Commun.*, vol. COM-30, Apr. 1982.

[37] P. C. Wong, "Pipeline rings and integrated services rings," Ph.D. dissertation, Dep. Electron., Chinese Univ. of Hong Kong, 1989.

Po-Choi Wong (S'82-M'89) was born in Hong Kong in 1958. He received the B.Sc. and M.Phil. degrees in electronics from the Chinese University of Hong Kong in 1983 and 1985, respectively, and is expecting to receive the Ph.D. degree in 1989.

Since July 1985, he has been a Lecturer in the Department of Computing Studies, Hong Kong Polytechnic, doing research and teaching in computer communications and networking. His current research interests are integrated services networks, computer systems performance, evaluation and office automation.

Tak-Shing Peter Yum (S'76-M'78—SM'86) was born in Shanghai, China, on February 26, 1953. He received the B.S., M.S., and Ph.D. degrees from Columbia University, New York, NY, in 1974, 1975, and 1978, respectively.

From 1978 to 1980, he was a member of the Technical Staff at Bell Laboratories, Holmdel, NJ, and worked on the performance analysis of the common channel interoffice signaling network. From September 1980 to June 1982, he was an Associate Professor at the Institute of Computer Engineering, National Chiao-Tung University, Taiwan. From August 1982 to July 1988 he was a Lecturer in the Department of Electronics, the Chinese University of Hong Kong. At present, he is a Senior Lecturer in the Department of Information Engineering at the same university. His current research interest is in the design and management of future information networks.