A Distributed, Fair, and Efficient Protocol for Integrated Voice/Data Services on Token Ring Networks

Jau-Hsiung Huang, Chi-Wen Chen, and Ming-Cheng Lee

Department of Computer Science and Information Engineering

National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

This paper proposes a protocol which supports integrated services on a token ring network. The key features of the protocol follow. (1) Both synchronous and asynchronous traffics, such as voice and data, are supported. (2) The control station of the protocol is not assigned to a fixed station. By so doing, this protocol is more distributedly controlled. (3) All voice stations have a fair chance of accessing the channel. In some other works, stations closer to the control station has a higher chance of accessing the channel than stations farther away from the control station. This situation will not happen in our protocol. (4) Data traffic has a higher chance of accessing the channel without influencing the quality of the synchronous traffic. In our protocol, data traffic does not have to wait to be transmitted until all synchronous traffic are sent. (5) Expedited data traffic is supported. An analytic model and simulations are included in this paper to show that this protocol is robust, efficient, fair, and presents small delay for data traffic.

1. Introduction

Different types of traffic have different transmission requirements. For example, voice packets are generated at a constant rate (normally at 64kbps using PCM encoding technique) while a person is talking and, for the integrity of the conversation, these packets must be delivered within a maximum delay (normally ranges from 10ms to 600ms) [1]. Furthermore, voice packets can tolerate a 1-2% of packet lost probability without seriously degrading the quality of the conversation [2],[3],[4]. Meanwhile, occasionally preventing a new call from successfully connecting (i.e., blocking the call) is acceptable. Once a call is established, however, this connection should not be aborted until the conversation finishes.

Data traffic, on the other hand, must be transmitted without being lost and without error. Moreover, data traffic is bursty and may demand a very high bandwidth occasionally. However, a random delay of data packets does not usually present a problem.

This project is sponsered by ALCATEL / TAISEL

For integrated voice/data token ring networks, there is usually a control station which takes charge of the voice connection [5][6]. That is, when a new call is to be established, the caller must first contact the control station and the control station will determine whether this connection should be allowed or not according to the present voice traffic. The disadvantage of this scheme is that the control is not fully distributed and requires operation overhead. Further, the crash of the control station may immediately crashes the ring.

Another way of implementing integrated voice/data token ring also requires a control station which will determine whether the ring should now be operating in the voice subcycle or in the data subcycle [6]. It then give out a predefined number of free tokens to other stations. Stations wanting to transmit voice/data packets will have to receive a free token before sending. Under such an arrangement, stations closer to the control station, they may have a better chance of accessing the channel for voice traffic and data traffic since they can receive and use the free token earlier. This implies that in sending the synchronous traffic, stations farther away from the control station may not be able to access the free token before the synchronous traffic subcycle ends; hence, their voice quality is worse than those closer to the control station. This presents a fairness issue.

In this paper, a new protocol is proposed which does not assign the control station to a fixed station. Instead, each station in the ring will be the control station in turn and controls the ring operation for a short period of time. This makes the protocol a fully distributed one and all stations will have a fair chance of accessing the free token. This resolves the problem of robustness and fairness. Moreover, this protocol allows data traffic to have different priorities and the highest priority data traffic has the same priority as the synchronous traffic such that they do not have to wait until all synchronous traffic is sent. This characteristic shortens the system delay for the expedited data traffic.

2. Model and protocol descriptions

In this section, we first describe the model used in the paper. We then describe the packet format and the usages and meanings of each control field in the packet. The details of the protocol is also discussed in this section.

2.1 Model description

In this integrated voice/data token ring network, three types of traffics are supported. They are voice, expedited data, and normal data traffics. Expedited data packets always have a higher priority than normal data packets. For the real-time traffic, we guarantee that each voice station will receive a free voice token for transmitting voice packets within a predetermined time interval. We denote a *cycle* as this predetermined time interval which is also named as a *packetization period* for voice stations. In order to meet the real-time constraint for the voice traffic, a control station is required to monitor the cycle time. In this protocol, each station in the ring will be the control station in turn for one cycle time to achieve distributed control and fair access to the channel.

In this paper, we assume the cycle time to be 20 ms [7]. With 64kbps PCM encoding, each active voice station generates 1280 bits voice data during a cycle. Adding a token ring packet header of 168 bits [10] and an extra 18 control bits for this protocol, the size of a voice packet is 1466 bits. If we consider a 1000-station ring on a 2 km cable operating at 10 M bits/s, then the total ring latency is the sum of the round-trip propagation delay and the 1 bit delay latency for each station, which is 100+1000 = 1100 bits. We define K to be the number of slots in a cycle where each slot can accommodate one voice packet plus the packet overhead. Therefore, the total number of bits transmitted in 20 ms is equal to (K)(slot size in bits) + (ring latency in bits), or 200000 = K * 1466 + 1100, K is 135 therefore. For the rest of the paper, we define the following notation which are used in the protocol.

Nv : maximum number of voice slots in a cycle.

THac : maximum number of active voice stations allowed in the system.

COUNTv: number of voice packets sent in a cycle recorded by each station.

COUNTd: number of data packets sent in a cycle recorded by each station.

2.2 Packet format

In order to explain the details of the protocol, we first describe the packet format as shown in Figure 1.

Bits	1	1	10	2	2	1	1	
	P	STOP	AC	TYPE	R	B/F	D/V	DATA

Figure 1. The packet format

P (1 bit) : set by new control station at the beginning of a cycle and reset when the control station is giving out control to the next station

STOP(1 bit) : set to "1" by the control station to indicate that current cycle has ended and the station possessing the token should give back the free token to the control station.

AC (10 bits) : Records the number of active voice stations (up to 1024 active stations)

TYPE(2 bits) : Specifies the token type :

00 : (V/E) for voice and expedited data.

01 : (V) for voice only.

10 : (E/N) for expedited data and normal data.

R(2 bit) : These bits are used for reservation by expedited data only and are set by backlogged station(s) to indicate that there are station(s) waiting for free token to transmit expedited data. The details will be provided later.

B/F (1 bit) : Busy / Free token.

D/V (1 bit) : Specifies the packet type, "1" as voice packet , "0" as data packet.

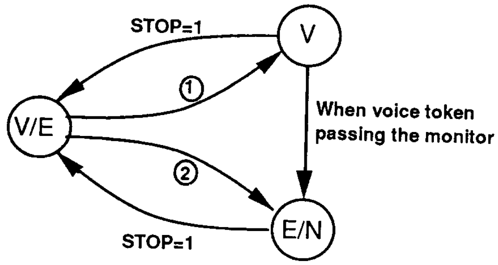
(This bit is used only when the packet is busy).

2.3 Protocol description

In this protocol, the control station is responsible for the synchronization for one cycle time. Each station in the ring will be the control station in turn for one cycle time. When the cycle time finishes, the control station will immediately regain the next free token to finish the cycle by setting the STOP bit to 1 in the passing by packet. The station sending the packet will generate a free token with STOP=1 which signifies to all other stations that this free token can only be used by the control station. When the current control station receives the free token, it finishes the current cycle and gives out the control to the next station by setting the P bit to 0 in the free token. A station sees a packet with P=0 means that it is the new control station. It then sets P to be 1 and starts a new cycle.

For expedited data, we wish that they can be sent as soon as possible and do not have to wait until all voice stations finish sending their voice packets; hence, we give the same priority to the expedited data and voice as long as the real-time constraint of the voice data can be fulfilled. To achieve this goal, the token mode has three kinds of access type, for voice/expedited data (V/E), for voice only (V) and for expedited/normal data (E/N). To begin a cycle, the control station initializes the token with type=(V/E), which allows stations to send voice and expedited data. When transmitted voice packet number is equal to maximum number of voice slots in a cycle (Nv) or control station receives a free voice token (V/E or V), which means that all voice stations have finished sending their voice packets, the token mode is changed to data mode(E/N). Figure 2 shows the flow of the protocol.

However, for maintaining a high quality of voice traffic, we should prevent the expedited data from occupying too many slots in V/E mode. If the number of data packets sent is equal to K-Nv, which is the minimum number of data slots in a cycle, in the V/E mode, the station that possesses the token should change the token mode to be for voice only(V). Every station counts the voice and data packets number by checking the V/D bit of passing packets and stores them in COUNTv and COUNTd respectively. Every station could know the start of a new cycle by a transition of token type from V or E/N to



① COUNTd = K-Nv

② COUNTv = Nv or
voice token passing the monitor

Figure 2. Three states of token accessing type

V/E. When a cycle starts, every station resets COUNTv and COUNTd to 0.

This integrated network controls the number of active voice stations by recording the number of active stations in the AC field in the token. When a station receives a new call, the station waits until it grabs a free voice token and check if the AC counter is smaller than the maximum active stations number, THac. If so, then the call is permitted and the value of AC is increased by 1. Otherwise, the call is blocked. Similarly, when a call is to be disconnected, the station grabs a free voice token, sends a disconnection packet to destination, and decreases the AC by 1.

When the network is operated in the E/N data mode, expedited data has a higher priority than the normal data. There are two bits in the R field. First R bit is the reservation bit used for the first expedited station that sees this packet. Second R bit is used for other stations with expedited data. If a station have expedited data to send, it can reserve the next slot by setting R bits in the following way. When a packet passing by, it examines the R bits. If the first R bit equals 0, then it sets the first R bit to 1 and clears the second R bit to 0. By so doing, other stations with expedited data can make further reservation by setting the second R bit. The station to generate the next free token examines the R bits to recognize the reservation of expedited data. It examines the first R bit when it is sending normal data, and examines the second R bit when it is sending expedited data respectively. If R bit set, it sends a free token with R bits equal to 01 to show that the slot is reserved for expedited data. The details of the action each station should take is included in the Appendix.

In the following we will present the mechanism to detect the crash of the control station. Note that in normal operation, a cycle begins from V/E type and ends in either V type or E/N type. Hence, if a station notices that there is no mode change in 135 slots, then the control station must have been crashed. Hence, the first station noticing the crash of the control station will become the new control station and starts a new cycle.

3. Performance evaluation of the protocol

In this section, we evaluate the performance of the protocol applying the Markov chain technique. We will first find the probability of voice packet lost when the number of active stations in the system is fixed. We then find the probability of voice packet lost in the average case. That is, we do not assume the number of active stations in the system is fixed.

3.1 Probability of voice packet lost with fixed number of active stations

Let Na be the number of active voice stations, and Nv be the allocated voice slots number in a cycle. If Nv is greater than Na then there will be no voice packet lost. However, in order to make good use of the silent interval of an active station, Nv could be chosen to be much smaller than Na [8], and the probability of voice packet lost could be calculated by the following analysis. Assume that the duration of talkspurts intervals and silence intervals are exponentially distributed with means $1/\mu=0.17$ and $1/\nu=0.41$s, respectively [9]. Then we have the Markov chain as shown in Figure 3 where the state is defined as the number of stations in talkspurt status. This technique is borrowed from [5].

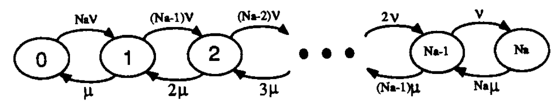


Figure 3. The Markov chain when there are N active stations in the system

$$p_n = p_{n-1}\frac{(Na - n + 1)\,\nu}{n\,\mu} = p_0 \prod_{i=1}^{n} \frac{(Na - i + 1)\,\nu}{i\mu}$$

$$p_0 = \frac{1}{1 + \sum_{n=1}^{Na} \prod_{i=1}^{n} \frac{(Na - i + 1)\,\nu}{i\mu}} = \frac{1}{1+\sum_{n=1}^{Na} \binom{Na}{n} r^n} = (1+r)^{-Na}$$

where $r=\frac{\nu}{\mu}$. From the above results, we have

P[voice packet lost | Na active stations]

$$= \frac{\text{Expected number of lost voice packets in a cycle}}{\text{Expected number of voice packets generated in a cycle}}$$

$$= \frac{\sum_{n=Nv+1}^{Na} (n - Nv)\, p_n}{\sum_{n=0}^{Na} n\, p_n} = \frac{\nu + \mu}{\nu\, Na} \sum_{n=Nv+1}^{Na} (n - Nv)p_n \qquad (1)$$

42.5.3.

From equation (1) we are able to achieve the voice packets lost probability versus the number of active stations as shown in Figure 4. Note that if Nv=134, we are able to support 380 active stations with voice packets lost probability less than 1%. Similarly, we can support 225 stations with less than 1% voice packets lost if Nv is set to be 80.

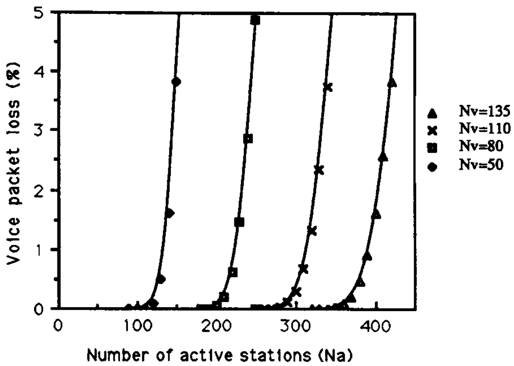


Figure 4. Probability of voice packets lost versus the active station number.

3.2 Probability of voice packet lost in the average case

In this section, we will find the distribution of active stations given that there are totally N stations in the system. Assume the call intervals (i.e., the station is active) and idle intervals of stations are exponentially distributed with mean 1/u=180 secs (3 mins) and 1/v=900 secs (15 mins), respectively. The system can be described by the follow birth-death process where the state is the number of active stations.

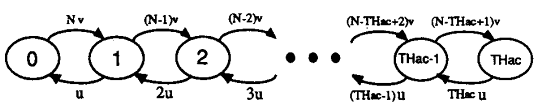


Figure 5. The Markov chain when at most THac active stations is allowed in the system.

$$(N-K)v\Pi_k = (k+1)u\Pi_{k+1} \qquad 0 \le k \le THac-1$$

$$\Pi_k = \binom{n}{k} \left(\frac{v}{u}\right)^k \Pi_0 \qquad k \le THac \qquad (2)$$

$$\sum_{k=0}^{THac} \binom{n}{k} \left(\frac{v}{u}\right)^k \Pi_0 = 1$$

$$\Pi_0 = \frac{1}{\sum_{k=0}^{THac} \binom{n}{k} \left(\frac{v}{u}\right)^k} \qquad (3)$$

From equations (1), (2) and (3), we can find the probability of lost voice packets given N stations in the system as shown below:

$$P[loss] = \sum_{n=0}^{N} P[loss \mid n \text{ active stations}] \cdot \Pi_n \qquad (4)$$

Figure 6 shows the probability of voice packets lost versus different values of THac derived from equation (4). It is interesting to note that there is a sharp rise around THac=226. Hence, we should be more conservative in choosing the value for THac to avoid the sharp increase in voice packets lost probability.

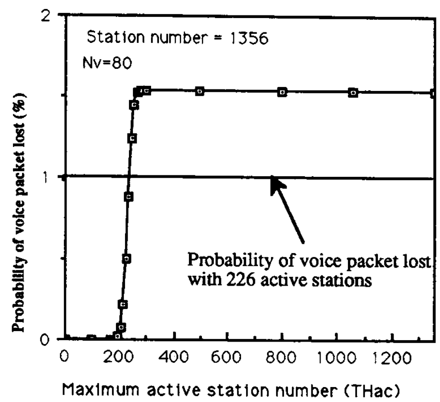


Figure 6. The probability of voice packets lost versus THac.

In this section, some simulation results are presented to show the performance of the protocol. In the simulation model, the following specifications are used. (1) All stations in the ring are active at all time and are uniformly distributed on the ring. (2) The distribution of the talkspurt duration is exponentially distributed with a mean interval of 0.17 seconds. Similarly, the silence interval is assumed to be exponentially distributed with 0.41 seconds mean interval. (3) The voice signal is sampled at 8 kHz and encoded into the 64 kbps PCM signal. (4) The packetization interval is chosen to be 20 ms.

Figure 7 shows that the delay for the expedited data is small (less than 1 cycle) in a heavily loaded system with Sv=0.4 and Sd=0.55 under various combinations of expedited data and normal data. We define IR to be the expedited data rate/ all data rate. We observe that the delay for the expedited data is very small until IR approaches 1.0 , i.e. almost all data is expedited. The small expedited data delay is due to high priority of expedited data. Certainly, the small delay of the expedited data is at the cost of the high delay of the normal data.

Figure 8 shows the data packet delay for both expedited data and normal data if the normal data rate is 9 times of the expedited data rate. Note that the delay of the expedited data packets is always less than the delay of normal data and is very small until the system is heavily saturated.

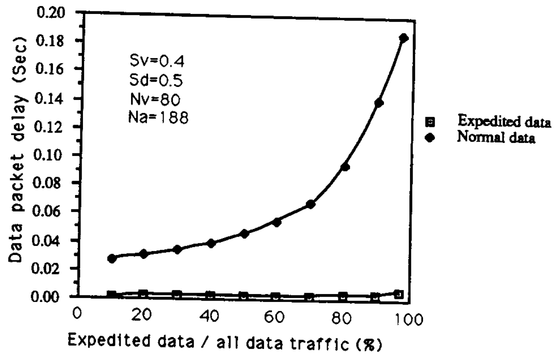


Figure 7. Mean delay for interactive data traffic.

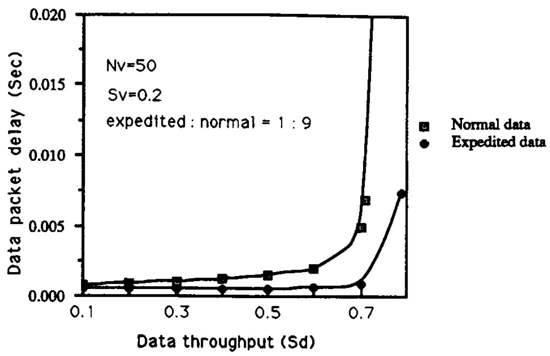


Figure 8. The mean data packets delay for Sv=0.2.

6. Conclusion

A protocol for integrated voice/data services on a token ring network is presented. Since we allow each station to be the control station for one cycle time, this makes this protocol a distributed and fair one. By using THac, we are able to control the quality of voice traffic when we allow much more call connections than Nv. Further, we allow the expedited data to have the same priority as the voice traffic and has a higher priority than normal data, this allows the delay of the expedited data to be very small. Lastly, the crash of the control station can easily be detected by other stations to make this protocol a robust one.

References

[1] P. T. Brady, "A Technique for Investigating On-Off Patterns of Speech," Bell System Tech. Journal, Vol.44, Jan 1965.

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

[3] S. R. Sachs, "Aliemative Local Area Network Access Protocols," IEEE Communications, Vol.26, pp.25-45, March 1988.

[4] N. S. Jayant and S.W. Christense, "Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure," IEEE Trans. Communication, Vol.COM-29, pp.101-109, Feburary 1981.

[5] Po-Choi Wang and Tak-Shing Peter Yum, "An Integrated Services Token-Controlled Ring Network," IEEE Journal on Selected Areas in Communications, Vol.7, No.5, pp.670-679, June 1989.

[6] John O. Limb and Lois E. Flamm, "A Distributed Local Area Network Packet Protocol for Combined Voice and Data Transmission," IEEE Journal on Selected Areas in Communications, Vol.SAC-1, NO.5, November 1983.

[7] Clifford J. Weinstein and James W. Forgie, "Experience with Speech Communication in Packet Networks," IEEE Journal on Selected Areas in Communications, Vol.SAC-1, NO.6, pp.963-978, December 1983.

[8] K. Bullington and J. M. Frazer, "Engineering Aspects of TASI," Bell System Tech. Journal, Vol.38, pp.353-364, 1959.

[9] J. G. Gruber. "A Comparison of Measured and Calculated Speech Temporal Parameters Relevant to Speech Activity Detection," IEEE Trans. Communications, Vol.COM-30, Apr, 1982.

[10] IEEE Standard 802.5-1985, "Local Area Network Standard: Tokenpassing Bus Access Methods and Physical Layer Specifications," 1985.