Decentralized token-CSMA/CD protocol for integrated voice/data LANs

Meng-Tsong Shieh*, Jang-Ping Sheu* and Wen-Tsuen Chen† propose a decentralized protocol for integrated voice/data local area networks

In this paper we propose a decentralized protocol for integrated voice and data local area networks. This protocol is based on a hybrid token-CSMA/CD protocol which combines all the merits of the CSMA/CD network and the token-passing network. We apply the hybrid protocol to voice and data individually, and offer two implicit tokens, one voice token and one data token, to achieve the priorities requirements. The voice transmission follows the token-passing protocol with a contention-resolution mechanism. It enforces a bounded delay to guarantee the real-time delivery for voice. This hybrid protocol will serve data of the low priority as fairly as best. To evaluate the protocol's performance, analytical formulations are derived. Numerical results are obtained for the throughput as well as the mean packet delay.

Keywords: local area networks, voice/data integration, token bus, hybrid token-CSMA/CD, performance analysis

INTRODUCTION

Integration of voice and data on local area networks (LAN) has recently become more and more important, particularly in office automation. The characteristics and requirements for their traffic is different[1,2]. Data traffic requires reliable delivery because one bit error may destroy the whole message. However, voice traffic is tolerant of some loss but requires real-time delivery. If both voice and data traffic is integrated into one network, the interaction between them may seriously affect the system behaviour. So, careful choice of a suitable LAN for integrated services is very important. Many random access protocols for integrated services have been proposed[3-6]. These protocols have an unbounded delay and variations of delay for delivery messages due to excessive collisions. This phenomenon causes random access protocols not to be suitable for voice communications, but with the token-passing protocols, the channel access delay is bounded. This makes the token-passing protocol more attractive for a mixed voice and data communication system. The integration of voice and data on a token ring has been studied for many years, both in the centralized system[9] and the decentralized system[10,11]. In the centralized ring, there exists a centre station to control the priorities accesses. In the decentralized ring, they implement the access priorities by use of a priority field and a reservation field. The main demerit is the poor reliability, such as the token loss, the duplicate token and the failure station, etc.

The current trend is to design a decentralized token bus protocol with high reliability. Several researchers proposed the scheduled bus protocols for integrated communication[7,8,12-14]. However, these protocols have

*Department of Electrical Engineering, National Central University, Chung-Li 32054, Taiwan, Republic of China
†Institute of Computer Science, National Tsing-Hua University Hsing-Chu 30043, Taiwan, Republic of China
Paper received: 2 June 1990. Revised paper received: 26 November 1990

0140-3664/91/004223-12

complex scheduling mechanisms and need complex reinitialization procedures at adding or deleting a station. Such complexities may decrease their reliability. The main disadvantage of general token-passing bus protocols is that the performance is worse than the random access protocols at light load. Wong and Gopal[15] have proposed a hybrid token-CSMA/CD protocol which performs as well as CSMA/CD at light load and as well as a token-passing protocol at heavy load. This hybrid protocol is suitable for real-time communications.

In this paper we propose a hybrid protocol that can be used for the integration of voice and data on a bus network. There are two implicit tokens on the bus to implement the priorities requirements. The voice token has higher transmission priority than the data one. The data transmission protocol is a hybrid token-CSMA/CD protocol which serves the data well at any time. The protocol is described, and a model of the protocol and performance analysis is given. Simulation and performance results are presented, and we finally give some conclusions.

TRANSMISSION PROTOCOL

The most important issue for integrated voice/data networks is that voice traffic must satisfy real-time requirements. This can be accomplished by limiting data transmission to a level that will not seriously affect voice quality. Hence, a token-passing bus protocol is applied to voice traffic to guarantee a bounded delay. A low-priority hybrid token-CSMA/CD protocol will permit data traffic to use the remaining network capacity as much as possible.

Our network nodes are configured in a bus topology, as shown in Figure 1. The nodes are directly attached to the bus network, and each contains a node interface to manage the channel access control. Each node may be attached with a single voice station, a single data station, or even multiple voice and data stations. In other words, each node can handle both voice and data traffic. There are two originally addressed systems: the conversation system with a total of N_v voice stations addressed from 0 to $N_v - 1$, and the data communication system with a total of N_d data stations addressed from 0 to $N_d - 1$.

Basic definitions and assumptions

The major disadvantage of the token bus network architecture is the delay involved in passing the token through idle stations. This delay could be minimized by allowing the token to skip idle stations. This will reduce the voice delay and achieve real-time delivery for more calls. Therefore, we should apply the token-skipping scheme to voice transmission protocol. On the contrary, the data token circulates on the originally addressed data system. The reason why the data system does not employee the token-skipping scheme is discussed below.

In our protocol, two implicit tokens - voice token and data token - simultaneously circulate on the bus system at the same time. In our token-skipping scheme, the voice token is only rotating on the conversation system which consists of all active voice stations. A voice station is called in 'active' state when it communicates with another one. In the data system, the station with no packets to transmit is called in 'thinking' state, and will become 'backlogged' if it has a packet queued or undergoing transmission. When a voice call is established, these two communicating voice stations will be added to the conversation system, and be given a conversation number individually. When their communication terminates, they return their conversation numbers to the conversation system. Each time a voice station participates in the conversation system, it will be given a new conversation number depending on the total number of conversations in progress. Hence, the conversation system is call-oriented with a dynamic-numbering strategy.

Here, the channel is slotted and the slot size is the end-to-end propagation delay upon the overall bus system. All stations are assumed to be synchronized, and the transmission will start at the beginning of a slot. A token-holding time is the time that a station holds the token. The action of token-holding time is the time that a station holds the token. The action of token-holding depends on the event EOC (end of carrier), defined as the event of channel state changing from busy to idle. A token is passed at the end of a token-holding. After the channel state changes from busy to idle, there exists an idle slot as a gap to distinguish the two adjacent token-holdings. This idle slot assures that all attached stations can detect the action of token-passing synchronously.

Voice transmission protocol

The conversation system consisting of all active stations acts according to the usual token bus protocol. At the end of a token-holding, the voice station in possession of the voice token has the priority for transmission. When the voice token owner detects an EOC and has a packet for transmission, it will transmit its packet immediately. At the

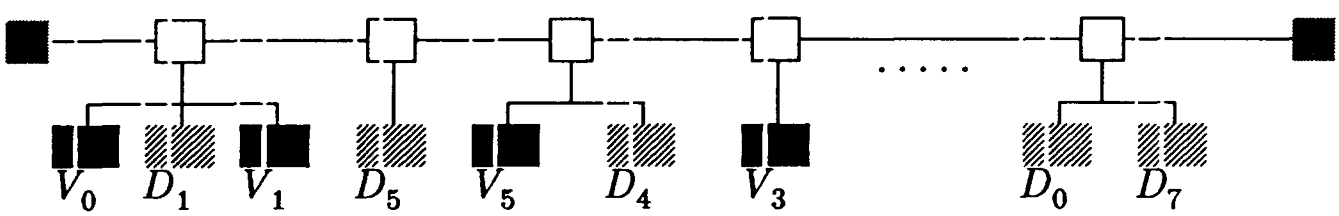

Figure 1. Integrated voice/data network model. V_i*: voice station numbered i;* D_j*: data station addressed j*

end of this transmission, the voice token is passes to its successor. If the voice token owner has no packet for transmission, it does nothing, and the channel remains idle. The backlogged data stations will find this idle slot and then get transmission opportunities. An active voice station can not transmit messages until it seizes the voice token.

Note that a voice station out of the conversation system may want to establish a new call with another voice station. They must request to participate in the conversation system first, before establishing a connection. The requestor will send a participate_request packet according to the CSMA/CD protocol. If a collision is detected, it terminates its participate_request packet and waits for the next idle slot. The voice token owner will retransmit immediately when it collides with a participate_request packet. In normal operation, no packet collisions of active voice stations happen. If the participate_request message suffers many collisions, it will denote that there are too many calls on the conversation system. The requestor will not successfully participate in the conversation system until some calls terminate. This procedure always provides information towards setting up a new call, while at heavy voice load it blocks further voice calls entering the conversation system.

Data transmission protocol

The data stations act according to the hybrid token-CSMA/CD protocol, and their actions depend on the total number of continuous idle slots at the beginning of a token-holding. Each time a token, voice or data, is passed, the backlogged data stations will not transmit their messages immediately but keep sensing the channel state. If the channel state is still idle after a slot, it denotes that there is no voice packet transmission. Upon detecting these two continuous idle slots, including an idle slot as a gap, all backlogged data stations except the data token owner will transmit their packets with probability v. If a collision is detected, all data stations terminate their transmission and wait for another two continuous idle slots to retransmit with probability v again.

However, the data station possessing the data token behaves in a different way. When the data token owner has detected two continuous idle slots, it has the priority to transmit its packet. In the event of a collision, the data token owner does not terminate its transmission. Specifically, it keeps the channel busy for a time interval not smaller than a slot, and then retransmits its packet immediately. During this collision slot, all other contending data stations terminate their transmission and wait for another two continuous idle slots.

We do not apply the token-skipping scheme to data system for two reasons. First, data systems follow a hybrid token-CSMA/CD protocol, not a usual token bus protocol. A data station not holding a token still has successful transmission opportunities by following v-persistent CSMA/CD. The idle data token owner will not affect the throughput seriously, even without the token-skipping scheme. Second, the token-skipping scheme needs complex procedures for initialization, station insertion and deletion, and call establishment and termination. For voice systems, a telephone call consists of a caller and a callee. However, a data communication may be a broadcast communication. The broadcast communication makes the dynamic-numbering strategy of token-skipping more and more complex, and complex procedures make fault-tolerance difficult. Correct delivery of packets is very important for data communication. Therefore, we do not apply the token-skipping scheme to the data communication system.

Action of implicit token-passing

The voice token is passed to the successor whenever an EOC is detected. The data token is passed only when an EOC is detected with two continuous idle slots ahead of a transmission. But, there still exists one problem. If there are no voice and data transmissions at a particular period, the EOC will not happen. It results in both tokens staying for a long time and causes an unbounded delay for the voice traffic. Hence, we should take some steps to limit the token-holding time in one station.

When *all* the data stations have detected three continuous idle slots, including an idle slot as a gap, they think both token owners idle and then issue dummy packets together to change the channel state. At the end of dummy packets, both tokens are successfully passed to their successors. The addition of dummy packets avoids the unacceptable delay due to the transient zero load and offers a synchronization at the same time. Thus, the data token is passed when an EOC with two or three idle slots ahead of transmission is detected. However, the voice token is passed whenever an EOC is detected.

It is obvious that the same voice station will seize the voice token again after N times of token-holdings, with N being the number of active voice stations. We define the voice token rotation time as the elapsed time from the instant a voice station possesses the voice token until the next instant the same station seizes the voice token. Clearly, the voice token rotation time is just the upper limit of voice packet delay, and it is composed of N times token-holdings. These N times token-holdings may be voice or data transmissions, collisions or dummy packet transmissions. Because the voice token rotation time depends on N, we develop a call-oriented protocol to reduce N as much as possible. We find that the amount of data load has little effect on the voice traffic.

Implementation

Note that the token passing is implicit, i.e. the action is caused by the EOC, and not the explicit transmission of a control message. To implement implicit token-passing, each station is required to keep track of some state registers. The implementation requires that each voice station should be equipped with four regesters, α, β, γ and δ. Their functions are described as follows:

α — records the number of voice conversations in progress;
β — each time an EOC is detected, the station increment β by 1 modulo 2α;
γ — the physical address in the originally assigned voice system;
δ — the current conversation number in the conversation system.

Each active voice station has its own conversation number stored in δ. When the system is initialized, all βs are zeros. Each time the channel state changes from busy to idle, they increment their βs by 1. A voice station possesses the voice token when the value of β modulo 2α is equal to its conversation number δ.

The implementation of the data system requires that each data station has three registers, U, X and Y:

U — records the address of the data station;
X — counts the number of continuous idle slots;
Y — all Ys are incremented by 1 modulo N_d after the channel state changes from busy to idle and the value of X equals 2 or 3.

The action of Us, Xs and Ys is similar to the voice system. When the channel state changes from busy to idle, the Xs beging to count, and will stop counting upon detecting a busy slot. When the value of X equals 2, the station with $U = Y$ possesses the data token. If the value of X is 3, all data stations transmit dummy packets simultaneously.

Call management

When one voice station out of the conversation system wants to establish a new call, it will request to participate in the conversation system at first. We call the requestor the *primary* station, and the one it wants to communicate with, the *secondary* station. The primary station will transmit a participate_request message according to CSMA/CD. When the participate_request message is transmitted successfully, the current voice token owner makes a decision to accept or reject this call. The voice token owner will compare the current value of α with a constant η, the upper bound of conversation capacity. If α is less then η, this new call is allowed to establish, and the voice token owner will increment its α by 1 and then transmits a participate_confirm message. Otherwise, this call is rejected and the voice token owner will transmit a participate_reject message. The participate_confirm message will tell all active voice stations to load their values of α with the new value. This message also tells the primary station to load its δ with $\delta_p = 2\alpha - 2$, as its conversation number, and to load its α with the same value of the other active stations. If there is no station on the conversation system or the voice token owner is a failure station, the primary station will not receive a participate_confirm message and will then time-out. After the first time-out, the primary station guesses that the voice token owner is a failure station and will request to participate again. If the second time-out happens, it guesses that there is no active station on the conversation system, because the probability of two failure stations is very small. Then, the primary station transmits a declaration_packet to tell all stations 'I am the first participating station'. If the voice token owner that the requestor asks secondly is still a failure station, there truly exists a station with conversation number 0 to tell the primary station 'You are wrong', and the primary station would try to participate again. If no station answers after one voice token rotation time, the requestor loads its α with 1 and δ with 0, and then becomes the first participating station. The initialization is completed.

After completing the participation of the primary station, the primary station transmits a connect_request message to its corresponding secondary station when it seizes the voice token. This message tells the secondary station to load its α with the same value of the other active stations and to load its δ with $\delta_s = \delta_p + 1$. If the secondary station has been on the conversation system, it will find duplicate value of δ and then transmit a connect_reject message when seizing the voice token. This message tells all active stations to decrement their αs by 1, and the primary station will be deleted from the conversation system. The stations with conversation number larger than δ_s will decrement their δs by 2. The rejection is completed. If the secondary station is out of the conversation system and permits this connection, it transmits a connect_confirm message while seizing the voice token. The participation of the secondary station is achieved. When two communicating stations, whose conversation numbers are m and $(m + 1)$, want to terminate their call, one of them will transmit an connect_end message and then all active stations decrement their αs by 1, and those stations with conversation numbers larger than $m + 1$ will decrement their δs by 2. We find that the communicating pair will participate in, or be deleted from, the conversation together, and have adjacent conversation numbers.

PERFORMANCE ANALYSIS

In this section our objective is to obtain accurate expressions for the mean voice delay, data delay and bandwidth allocation. We divide packet transmissions into five types of token-holding. Since a voice token rotation period is composed of N times diverse types of token-holdings, we should find the probability distribution function (pdf) of token-holdings by using an imbedded Markov chain. Based on the expression of the voice token rotation time, we can formulate the bandwidth allocation expression. Applying Little's theorem and combination mathematics to voice and data systems, respectively, we can compute all performance measures.

Basic definitions and assumptions

As described above, the voice token only circulates on the conversation system. Hence, we just take care of the conversation system and assume there are N active stations on the conversation system. Each voice or data station has a single buffer, but the action of the single buffer is different in each of them. In each active voice station, a new arrival will delete the previously queued packet and then occupies this buffer, i.e. the voice buffer queues the last arrival packet, and the previously generated packets are lost. However, each backlogged data station will generate a new packet only when the queued packet has been successfully transmitted. The single buffer inhibits a backlogged data station generating any new data packet. We assume that each thinking station will become a backlogged station in a slot with probability σ. Note that the data arrivals are not memoryless under this assumption.

The Poisson process is the arrival process used most frequently to model the behaviour of queues[16]. Three basic statements are used to define the Poisson arrival process. Consider a small time interval Δt ($\Delta t \rightarrow 0$), separating times t and $t + \Delta t$. Then:

1. The probability of one arrival in the interval Δt is defined to be $\lambda_d \Delta t + 0(\Delta t)$, with $\lambda_d \Delta t << 1$, and λ_d is a specified probability constant.
2. The probability of zero arrivals in Δt is $1 - \lambda_d \Delta t + 0(\Delta t)$.
3. Arrivals are memoryless: an arrival (event) in one time interval of length Δt is independent of events in previous or future intervals.

With the first two definitions of the Poisson process, the value of λ_d approaches σ if we treat the data arrival model as a rate of λ_d with Poisson distributions. The difference between the data arrival model and Poisson proces is point 3 above. A backlogged data station cannot generate any new packet under the preceding model with σ, i.e. an arrival is dependent on the previous arrival. Heavy data load makes a data station backlogged for a long time. Hence the value of σ should be as small as possible. In the following discussion, we let σ equal λ_d. These alternating models will simplify our performance analysis. Assume the mean packet transmission time is T for voice and data both. As shown in Figure 2, the token-holding can be classified into five types: V, D, C, F and Z.

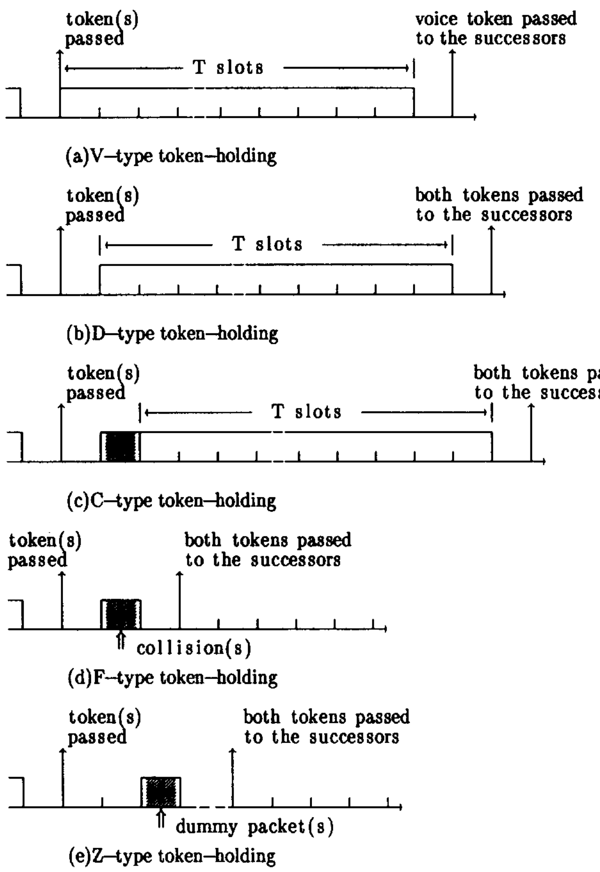

Figure 2. Five types of token-holdings (see text)

- V type – represents a successful voice transmission.
  The voice token owner is ready and the token-holding time is $T + 1$ slots.
- D type – represents a successful data transmission without a collision ahead.
  The voice token owner is not ready and there are two cases of the D type token-holding:
  (i) the data token owner is backlogged and the other backlogged stations do not transmit packets.
  (ii) the data token owner is thinking, but exactly one of the backlogged stations transmits its packet following v-persistent CSMA/CD.
  The token-holding time is $T + 2$ slots.
- C type – represents a successful data transmission with a collision ahead.
  The voice token owner is not ready. The data token owner is backlogged and at least one of the other backlogs transmits its packet simultaneously. The data token owner will retransmit successfully after a collision slot. The token-holding time is $T + 3$ slots.
- F type – represents a failure data transmission.
  Neither token owners have packets to send, and more than one backlogged data station transmit their packets simultaneously. The token-holding time is 3 slots.
- Z type – represents the condition of the dummy packets transmitted.
  The voice token owner is not ready, and no backlogged data station transmits its packet. The token-holding time is 4 slots.

Here we assume that the probability of a new call request is negligible. This assumption is reasonable, since the call interarrival times and durations are much larger than the mean voice token rotation time. Consequently, we have a fixed number of voice calls in each voice token rotation period. So, we do not consider the condition that the participate_request messages collide with the voice transmissions. Besides, the service is non-preemptive. Once the data station has begun to be served, a new arriving voice packet must wait for the next time it seizes the voice token. Thus, the voice token owner will not transmit its new packet after more than one idle slot. Based on the above definitions and assumptions, the results of the analysis are derived in the Appendix.

SIMULATION RESULTS

In our simulations, we assume that there exist 10 voice calls and 20 data stations, and both traffic types are assumed to be equally distributed among all stations. The packet transmission time is 100 slots for both traffic. For our bus network, the bus length is 3 Km and the transmission rate is 1 Mbit/s. This means that the slot size is 0.01 msec and the packet length is 1000 bit.

In Figure 3, we plot the characteristics of mean delay *versus* voice arrival rate λ, and the simulation results are compared with the analytic results discussed above. The main reason for the error between the analytical and simulation results is the offset of the initial estimate of R in equation (A5). In equation (A5), we have assumed that the probability of D-type token-holding equals that of the C-type. This assumption will lead to a larger error of R when the data load is very light or very heavy. When the data load is light, the number of collisions is few, and then the probability of C-type token-holding is less than that of the D-type. Under this condition, the accurate R will be smaller than the initial estimate of R obtained by equation (A5). The simulation results of mean delays would be larger than the analytical results of mean delays at light load because the voice stations will get the voice token again sooner. It seems that the error should get smaller, since the probability of C-type token-holding may be larger than that of D-type token-holding when the load becomes heavier. However, there still exists another factor to influence the mean voice packet delay. In equation (A5), we suppose that the system serves all packets fairly in one voice token rotation period. In fact, the number of lost voice packets increases as the total load increases. T_r in equation (A19), the residual life time, *does* take the lost packets into consideration, and the system still tries to transmit those packets that should be discarded[21]. Because the voice station only serves the last packet generated during one token rotation time, the accurate voice packet delay should be smaller than the analytical result resolved by equation (A19). Thus, we find that the simulation results are little smaller than the analytical results at heavy load.

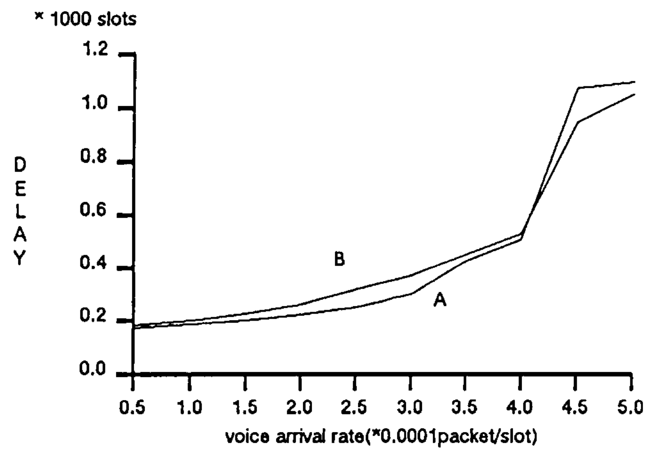

Figure 3. Mean delay versus λ *with* $\sigma = 0.0001$ *packet/ slot and* $v = 0.1$. *A: simulation results; B: analytic results*

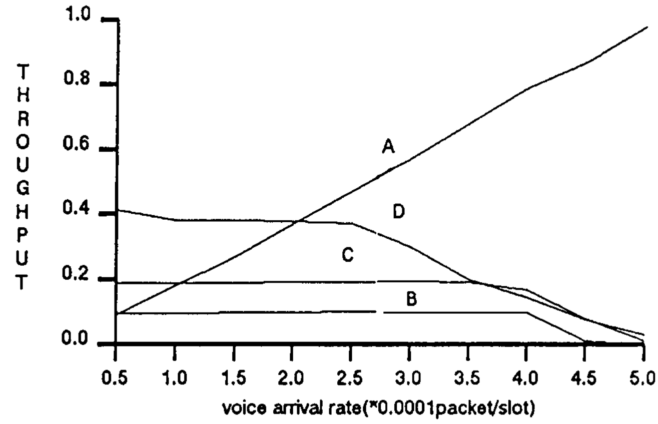

Figure 4. Throughput versus λ *for various* σ *with* $v = 0.1$. *A: voice for all* σ*s; B: data with* $\sigma = 0.00005$ *packet/slot; C: data with* $\sigma = 0.0001$ *packet/slot; D: data with* $\sigma = 0.0002$ *packet/slot*

Figure 4 shows the bandwidth in percentage plotted against λ for various values of σ. We see that the transmission bandwidth occupied by the voice traffic is little affected by the data traffic at light load. The bandwidth allocated to the voice traffic is almost proportional to λ when $(N\lambda R + N_d\sigma R) < N$. As described in the previous section, there are a number of token-holdings reserved for the data traffic, and the bandwidth of data is almost unchanged for a fixed σ under the condition $(N\lambda R + N_d\sigma R) < N$. It denotes that the excessive token-holdings will be fully utilized by the data traffic. When λ approaches those values resulting $(N\lambda R + N_d\sigma R) > N$, the bandwidth allocated to the voice is still proportional to λ, but the bandwidth allocated to the data will decrease. This is because the voice traffic with the high priority utilizes most token-holdings, and the number of excessive token-holdings will be insufficient for the data traffic. The instability will inhibit further data packet generations. As usual, we can use the TDM-like model to roughly estimate the allocated bandwidth. Consider the TDM with 20 transmission token-holdings per frame, and suppose the size of each transmission token-holdings is 100 slots (or a packet length). Therefore, the length of a frame is 2000 slots. Then, the bandwidth allocated to the voice traffic is about $200000\lambda\%$. The bandwidth for data is about $200000\sigma\%$ if $(N\lambda R + N_d\sigma R) < N$, and about $100(1 - \lambda R)\%$, otherwise. For example, the bandwidth occupied by the voice traffic is always about 80% when $\lambda = 0.0004$ packet/slot and the bandwidth for data is 20% when $\sigma = 0.0001$ packet/slot under the condition of $(N\lambda R + N_d\sigma R) < N$.

In Figures 5, 6 and 7, the mean delays are plotted as a function of λ for different values of σ. At light load, the voice message delay with high σ is larger than that with low σ. This is because the number of D-type and C-type token-holdings increases as σ increases. The mean voice packet should suffer a longer delay from more data packet transmissions. As λ increases, the voice packet delay also increases and finally converges to a limit. This limit is about half a voice token rotation time. The characteristics

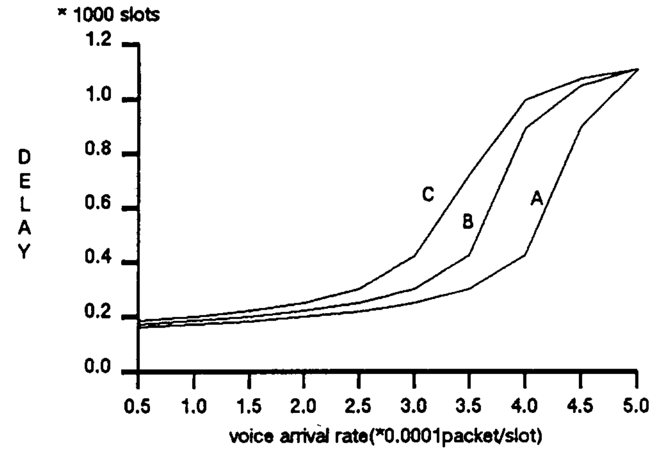

Figure 5. Voice packet delay versus λ for various σ with $v = 0.1$. A: $\sigma = 0.00005$ packet/slot; B: $\sigma = 0.0001$ packet/slot; C: $\sigma = 0.00015$ packet/slot

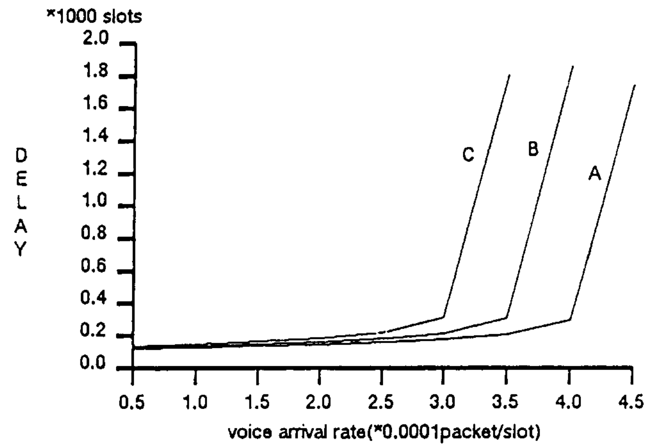

Figure 6. Data packet delay versus λ for various σ with v 0.1. A: $\sigma = 0.00005$ packet/slot; B: $\sigma = 0.0001$ packet/slot; C: $\sigma = 0.00015$ packet/slot

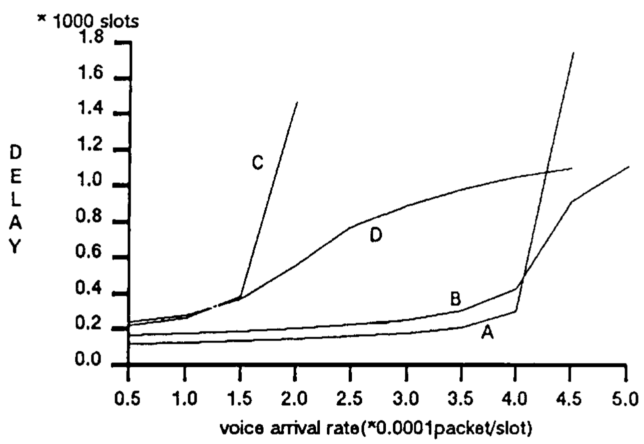

Figure 7. Mean delay versus λ for various σ with v 0.1. A: data with $\sigma = 0.0001$ packet/slot; B: voice with $\sigma = 0.0001$ packet/slot; C: data with $\sigma = 0.0003$ packet/slot; D: voice with $\sigma = 0.0003$ packet/slot

of voice packet delays is the same as the result of a usual token bus protocol. Remember that the voice traffic follows the token bus protocol, and the data traffic follows the hybrid token-CSMA/CD protocol. Because the performance of the CSMA/CD protocol is better than that of the token-passing protocol at light load, voice traffic suffers a longer delay than data traffic when the total load is light. As the total load increases, the mean data packet delay increases more rapidly than the voice packet delay. The phenomenon results from the reserved priority for voice in each token-holding. The system always serves the voice traffic first, and the data traffic is inhibited from further generations. Thus, the data delay increases rapidly to provide a bounded voice delay. When the total utilized bandwidth approaches 100%, the mean data packet diverges rapidly, and the mean voice packet delay converges.

In the preceding discussions, simulation runs are carried out for different voice arrival rates, which depend on voice digitization rate. In Figures 8 and 9, 64 Kbit/s

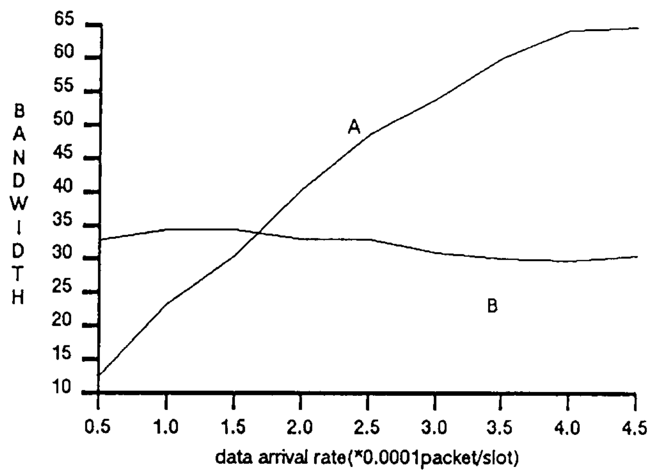

Figure 8. Effect of data load on bandwidth. A: bandwidth occupied by data; B: bandwidth occupied voice. (1_v = 1024 bits, r_v = 64 Kbit ADPCM)

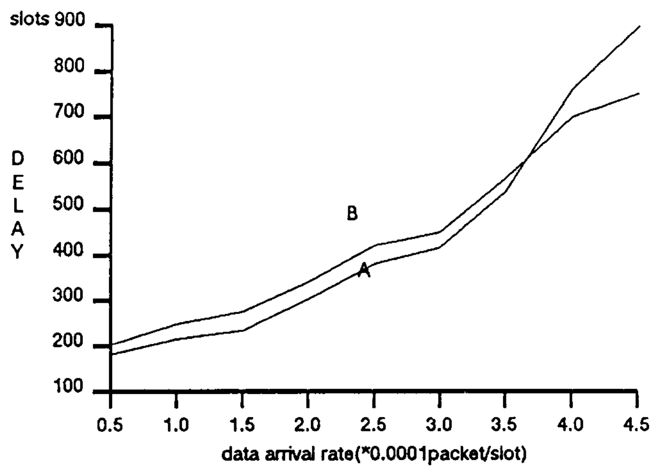

Figure 9. Comparison of voice and data delays for various data loads. A: data delay; B: voice delay. (1_v = 1024 bits, r_v = 64 Kbit ADPCM)

ADPCM speech coding with mean talkspurt duration of 350 ms and mean silence period fo 650 ms are assumed. By equations (A3) and (A4), the mean voice arrival rate will be 0.00017 packet/slot under this environment. In Figure 8 we investigate the effect of data traffic on the bandwidth occupied by voice. As the data load increases, the bandwidth for voice decreases only slightly. Obviously, a fixed bandwidth will be allocated to voice when the voice arrival rate is fixed. Figure 9 compares the delays in two systems, with various voice arrival rates. It can be seen that both delays increase as the data load increases. The voice delay is slightly larger than the data delay when the data load is light. As the data load increases, the increasing rate of data delay is larger than that of voice delay. For a high data load, the voice delay is still under control, but the data delay increases rapidly.

CONCLUSIONS

For a bus topology, a double-hybrid protocol is introduced in this paper. There exists one implicit voice token and one implicit data token to serve the voice traffic and the data traffic, respectively, and we let the voice token rotate with a higher priority. We apply the token-passing protocol to guarantee a bounded voice packet delay, and apply the hybrid token-CSMA/CD to serve data as well as possible. The maximum voice packet delay, or the voice token rotation time, depends on the total number of conversations. Hence, we propose the call-oriented strategy with dynamic-numbering to reduce the upper bound of the voice packet delay as much as possible. On the other hand, the silence interval of voice calls may be sufficiently utilized by the data traffic which follows the high performance hybrid token-CSMA/CD protocol.

However, the data traffic works worse due to the low priority at heavy voice load. If we want to reduce the voice packet delay further, we can also let the voice traffic follow the hybrid token-CSMA/CD protocol. This modified protocol will lead to a smaller voice packet delay and poor performance for data traffic. Recently (owing to being efficiently employed to carry a broad mixture of traffic in a wide range of application areas), the ability of LANs has become more and more important. Our protocol can easily be extended to such applications. The modified protocol may be called a 'multi-hybrid' protocol. We apply the alternating priorities mechanism to traffic with the real-time constraint and class services as introduced in the IEEE 802.5 token bus protocol to that traffic without the real-time constraint.

ACKNOWLEDGEMENT

This research was supported by Taiwan International Standard Electronics Ltd.

REFERENCES

1 **Forgie, J W** 'Speech transmission in packet-switched store-and-forward networks' *AFIPS, NCC* (1975) pp 137–142
2 **Schwartz, M** *Information Transmission, Modulation and Noise* (3rd ed.) McGraw-Hill, New York, USA (1980)
3 **Nutt, G T and Bayer, D L** 'Performance of CSMA/CD networks under combined voice and data loads' *IEEE Trans. Commun.* Vol 30 No 1 (January 1982) pp 7–11
4 **Gonsalves T A** 'Packet voice communication on an Ethernet local computer communication systems' *Proc. ACM SIGCOM '83* (1983) pp 178–185
5 **De Treville, J and Sincoskie, W D** 'A distributed experimental communication system' *IEEE J. Select Areas in Commun.* Vol 1 No 12 (December 1983) pp 1070–1075
6 **Kleinrock, L and Tobagi, F A** 'Packet switching in radio channels: Part I - Carrier sense multiple-access modes and their throughput-delay characteristics' *IEEE Trans. Commun.* Vol 28 (April 1980) pp 468–488
7 **Li, S Q and Majithia, J C** 'Performance analysis of a DTDMA local area network for voice and data' *Computer Networks* Vol 8 (1984) pp 81–91
8 **Goel, R K and Elkakeem, A K** 'A hybrid FARA/CSMA-CD protocol for voice-data integration' *Computer Networks & ISDN Syst.* (September 1985) pp 223–240
9 **Wong, J W and Gopal, P M** 'Analysis of a token ring protocol for voice transmission' *Computer Networks* Vol 8 (1984) pp 339–346
10 **Ibe, O C and Gibson, D T** 'Protocols for voice and data local area networks' *IEEE Commun. Magazine* Vol 24 (July 1986) pp 30–36
11 **Yu, J H and Chen, W J** 'Design consideration of an integrated voice/data token ring local area network' *Proc. ICS. 86* Taiwan, ROC (December 1986) pp 1791–1800
12 **Liu, T T, Li, L and Franta, W R** 'A decentralized conflict-free protocol, GBRAM for large scale local networks' *Proc. Comput. Network Symp.* (December 1981) pp 39–54
13 **Rios, M and Georganas, N D** 'A hybrid multiple access protocol for data and voice-packet over local area networks' *IEEE Trans. Comput.* Vol 34 (January 1985) pp 90–94
14 **Fine, M and Tobagi, F A** 'Demand assignment multiple access schemes in broadcast bus local area networks' *IEEE Trans. Comput.* Vol 33 (December 1984) pp 1130–1159
15 **Gopal, P M and Wong, J W** 'Analysis of a hybrid token-CSMA/CD protocol for bus networks' *Computer Networks & ISDN Syst.* (September 1985) pp 131–141
16 **Kleinrock, L** *Queueing systems: Volume I, Theory* John Wiley, New York, USA (1975)
17 **Tobagi, F A and Kleinrock, L** 'Packet switching in radio channels: Part IV - stability considerations and dynamic control in CSMA/CD' *IEEE Trans. Commun.* Vol 25 (October 1977) pp 1103–1120

APPENDIX 1

Voice model description

Figure 1 demonstrates the connection of voice stations to the network. In each voice station, the speech signal is digitized at rate r_v bits/s and segmented into packets of fixed-length l_v bits that correspond to a segment of speech of duration $P_v = l_v/r_v$ seconds. A speech activity device is present at each voice station such that silence periods in the speech signal are not transmitted. The voice packet delay will have three components, the time required to assemble a complete packet (i.e. P_v), the waiting time experienced in gaining access to the bus, and the packet transmission time. The bus network is configured so that the waiting time in gaining access to the bus is bounded above by P_v. The maximum possible total delay in delivering a packet is thus bounded by $2P_v$ plus a packet transmission, so in the worse case voice packet can be delivered synchronously with this delay. The packet generation time is independent of the packet transmission process; thus, the queue size at the buffer continues to grow while a packet is waiting for transmission. Since we assume single buffer capacity, there is some packet loss at the voice-source station. The loss rate should be less than 1-2%.

A bound on the maximum voice token rotation time, and hence the voice packet delay, is imposed by limiting the number of active voice stations N, since one voice token rotation time is composed of N times token-holding. The maximum possible voice token rotation time is less than a packetization period if:

$$N(T + 3) < l_v/r_v = P_v \tag{A1}$$

where $(T + 3)$ is the longest token-holding time. Dividing both sides by $(T + 3)$, we obtain the following:

$$N < \frac{l_v}{r_v(T + 3)} = \eta \tag{A2}$$

η is the maximum number of voice calls that can be simultaneously handled by the network subject to the bounded token rotation time constraint. It is determined by the voice coding and transmission rates.

Modelling the voice packet arrival process is the difficulty in this performance analysis. As long as the voice packet length is significantly less than the talkspurt lengths, the voice packet arrivals consist of periods during which packets arrive in deterministic fashion separated by periods of silence. The voice packet arrivals consist of the deterministic arrival of a geometrically distributed number of packets, followed by silence during an exponentially distributed time, plus a packetization period. Usually, the voice arrival process was modelled by the Poisson arrival of batches of voice packets. The expression for the mean burst length $E[B_2]$ and the mean batch arrival rate A_2^b is given by:

$$b_2 = B_2 = \frac{T_a^2 + 2T_aT_s + 2T_aT_s^2/P_v}{2(T_a + T_s)^2} \tag{A3}$$

and:

$$A_2^b = \frac{T_a}{b_2P_v(T_a + T_s)} \tag{A4}$$

where T_a is the mean talkspurt period and T_s is the mean silence period. For simplicity, we assume a stream of voice packets arrives at each active station at an average rate of $\lambda = A_2^b b_2$ with independent Poisson distributions.

Analysis using an imbedded Markov chain

Let R be the mean of the voice token rotation time. Since one voice token rotation time consists of N times those five types of token-holding, $N\lambda R$ and $N_d\sigma R$ will be the mean number of voice and data arrivals, respectively, during one voice token rotation time. Hence, there exist $(N\lambda R + N_d\sigma R)$ times of successful transmission and $(N - N\lambda R - N_d\sigma R)$ times of unsuccessful transmissions if all station queues are stable. The token-holding time may be $T + 2$ or $T + 3$ slots for data transmissions and be 3 or 4 slots for unsuccessful transmissions. For the sake of simplification, we assume the probability of D-type token-holdings equals that of C-type ones, and the probability of F-type token-holding equals that of Z-type ones. Then, we can obtain a function of R as follows:

$$R = N\lambda R(T + 1) + N_d\sigma R\frac{(T + 2) + (T + 3)}{2} + (N - N\lambda R - N_d\sigma R)\frac{3 + 4}{2} \tag{A5}$$

If $(N\lambda R + N_d\sigma R) > N$, then:

$$R = N\lambda R(T + 1) + (N - N\lambda R)\frac{(T + 2) + (T + 3)}{2} \tag{A6}$$

It is obvious that the overall system is stable only if $N\lambda R$ is less than N. When $(N\lambda R + N_d\sigma R) > N$, the data backlogged stations will inhibit further packet generations and the system allocates $(N - N\lambda R)$ times of token-holdings to the data traffic at most. This condition of overload should be avoided as possible, because it makes the alternating models of data arrival more unreliable. This inequality $(N\lambda R + N_d\sigma R) < N$, assures that all the packets can be served during a finite time. From equation (A5), we can resolve R:

$$R = \frac{7N/2}{1 - N\lambda(T - 5/2) - N_d\sigma(T - 1)} \tag{A7}$$

Having such an estimation of R, we can calculate the probability of each type of token-holding. Let us use $P_v(k)$, $P_d(k)$, $P_c(k)$, $P_f(k)$ and $P_z(k)$ as the probability of V-type, D-type, C-type, F-type and Z-type of token-holding, respectively, when the number of backlogged data stations at the beginning of a token-holding is k. We calculate these five probabilities as follows:

$$P_v(k) = \lambda R;$$

$$P_d(k) = (1 - \lambda R)\left[\frac{k}{N_d}(1 - v)^{k-1} + \left(1 - \frac{k}{N_d}\right)kv(1 - v)^{k-1}\right];$$

$$P_c(k) = (1 - \lambda R)\frac{k}{N_d}[1 - (1 - v)^{k-1}];$$

$$P_f(k) = (1 - \lambda R)\left(1 - \frac{k}{N_d}\right)[1 - (1 - v)^k - kv(1 - v)^{k-1}];$$

$$P_z(k) = (1 - \lambda R)\left(1 - \frac{k}{N_d}\right)(1 - v)^k \tag{A8}$$

λR is the probability that the voice token owner is ready and $\frac{k}{N_d}$ is the probability that the data token owner is among the backlogged stations. For the convenience of analysis, we define the elements of matrices V, D, C, F, Z, Q and H as follows:

$$q_{kj} = \begin{cases} 0, & j < k \\ \binom{N_d - k}{j - k}\sigma^{j-k}[1 - \sigma]^{N_d - j}, & j >= k \end{cases}$$

$$v_{kj} = \begin{cases} 0, & j < k \\ P_v(k) \cdot q_{kj}, & \text{otherwise} \end{cases}$$

$$d_{kj} = \begin{cases} 0, & j < k, \quad \text{or } k = 0 \\ P_d(k) \cdot q_{kj}, & \text{otherwise} \end{cases}$$

$$c_{kj} = \begin{cases} 0, & j < k, \quad \text{or } k < 2 \\ P_c(k) \cdot q_{kj}, & \text{otherwise} \end{cases}$$

$$f_{kj} = \begin{cases} 0, & j < k, \quad \text{or } k < 2 \\ P_f(k) \cdot q_{kj}, & \text{otherwise} \end{cases}$$

$$z_{kj} = \begin{cases} 0, & j < k \\ P_z(k) \cdot q_{kj}, & \text{otherwise} \end{cases}$$

$$h_{kj} = \begin{cases} 1, & j = k - 1 \\ 0, & \text{otherwise} \end{cases} \tag{A9}$$

The matrix Q represents the increase in backlog due to some of the thinking data stations becoming backlogged on finding the channel busy. q_{kj} denotes the probability that the number of backlogged data station changes from k to j in a slot. The matrix H represents the decrease in backlog after a successful data transmission (at this instant the backlogged data station re-enters the thinking mode). The slot transition matrices V, D, C, F and Z represent the increase in backlog at the beginning of V-type, D-type, C-type, F-type and Z-type token-holding, respectively. Their elements can be explained as follows.

All elements are 0 for $k > j$ because the number of backlogged data stations cannot decrease when the channel state changes from idle to busy. The matrix V deals with the V-type token-holding when the voice token owner is ready. It is obvious that v_{kj} is just equal to $P_v(k)q_{kj}$. The matrix D deals with the D-type token-holding when the voice token owner is not ready. $d_{kj} = 0$ for $k = 0$ because no backlogged station exist. There are two cases for $j > k$. One case is that the data token owner is backlogged, and its conditional probability is $\left[\frac{k}{N_d}(1 - v)^{k-1}\right]$. The other case is that the data token owner is thinking and exact one of the k backlogs transmits with probability v. The conditional probability of this case is $\left[\left(1 - \frac{k}{N_d}\right)kv(1 - v)^{k-1}\right]$. The matrices C, F and Z deal with the C-type, F-type and Z-type token-holding, respectively. Similar explanations can be provided for them. One case worth mentioning is that c_{kj} and f_{kj} are all 0 for $k < 2$. This is because no collision will happen if less than two backlogged data stations exist.

Let p_{ij} denote the probability that there are j backlogged data stations at the end of a token-holding, given that there are i backlogged stations at the beginning of the same token-holding. Then, the transition matrix P can be expressed as:

$$P = VQ^T + QDQ^TH + QCQ^{T+1}H + QFQ + QZQ^2 \tag{A10}$$

$[Q^k]_{ij}$ denotes the probability that the number of backlogged data stations changes from i and j after k consecutive slots. The right side of equation (A10) consists of five items. Each item represents the change in backlog during the corresponding type of token-holding. Take the first item, for example. Because the transition matrix V represents the increase in backlog at the beginning of a voice transmission. $[VQ^T]_{ij}$ denotes the increase in backlog after the V-type token-holding. The first Q of QDQ^TH represents the increase in backlog during the first idle slot of the D-type token-holding. Similar explanations are applied to the other items. Note that H of QDQ^TH and $QCQ^{T+1}H$ represents the decrease in backlog after a successful data transmission.

Let $\theta = [\theta_0\theta_1\theta_2 \cdots \cdot \theta_{N_d}]$ denote the steady-state probabilities of backlogged data stations at the beginning of a token-holding. These steady-state probabilities can be obtained by solving the system of equation $\theta = \theta P$ with the normalizing condition:

$$\sum_{i=0}^{N_d} \theta_i = 1 \tag{A11}$$

Performance analysis for data

As in Tobagi and Kleinrock[17], the average stationary channel throughput for data and voice, S_d and S_v, can be given by:

$$S_d = \left\{\sum_{k=0}^{N_d} \theta_k[P_d(k) + P_c(k)]T\right\} /$$

$$S_d = \left\{\sum_{k=0}^{N_d} \theta_k[(T + 1)P_v(k) + (T + 2)P_d(k) + (T + 3)P_c(k) + 3P_f(k) + 4P_z(k)]\right\};$$

and:

$$S_v = \left\{\sum_{k=0}^{N_d} \theta_k P_v(k) T\right\} /$$

$$S_d = \left\{ \sum_{k=0}^{N_d} \theta_k [(T+1)P_v(k) + (T+2)P_d(k) + (T+3)P_c(k) + 3P_f(k) + 4P_z(k)] \right\} \quad \text{(A12)}$$

Let $A(k)$ be the expected sum of backlogs over all slot in the busy period, with the number of backlogs being k at the beginning of this token-holding:

$$A(k) = \sum_{l=0}^{T} \left(\sum_{j=k}^{N_d} j[VQ^l]_{kj} \right) + \sum_{l=0}^{T} \left(\sum_{j=k}^{N_d} j[DQ^l]_{kj} \right) + \sum_{l=0}^{T+1} \left(\sum_{j=k}^{N_d} j[CQ^l]_{kj} \right) + \sum_{j=k}^{N_d} j[FQ^l]_{kj} + \sum_{j=k}^{N_d} j[ZQ^l]_{kj} \quad \text{(A13)}$$

By the use of $A(k)$, we obtain the average number of backlogged data stations $\overline{M}$ by:

$$\overline{M} = \left\{ \sum_{k=0}^{N_d} \theta_k A(k) \right\} \Big/ \left\{ \sum_{k=0}^{N_d} \theta_k [(T+1)P_v(k) + (T+2)P_d(k) + (T+3)P_c(k) + 3P_f(k) + 4P_z(k)] \right\} \quad \text{(A14)}$$

The average data packet delay $\overline{D}_d$ (normalized with respect to packet transmission time T) can be found by an invocation of Little's theorem in the following equation:

$$\overline{D}_d = \frac{\overline{M}}{S_d} \quad \text{(A15)}$$

Performance analysis for voice

We use another method to analyse the voice traffic. At first, we should calculate the mean probability for each type of token-holding by the following definitions:

$$\overline{P}_v = \sum_{k=0}^{N_d} \theta_k P_v(k);$$

$$\overline{P}_d = \sum_{k=0}^{N_d} \theta_k P_d(k);$$

$$\overline{P}_c = \sum_{k=0}^{N_d} \theta_k P_c(k);$$

$$\overline{P}_f = \sum_{k=0}^{N_d} \theta_k P_f(k);$$

$$\overline{P}_z = \sum_{k=0}^{N_d} \theta_k P_z(k) \quad \text{(A16)}$$

Since one voice token rotation time R is composed of N times token-holdings, R can be written in the following form:

$$R = \sum_{j_v=0}^{N} \sum_{j_d=0}^{N} \sum_{j_c=0}^{N} \sum_{j_f=0}^{N} \left\{ \frac{N!}{j_v! j_d! j_c! j_f! (N - j_v - j_d - j_c - j_f)!} \overline{P}_v^{j_v} \overline{P}_d^{j_d} \overline{P}_c^{j_c} \overline{P}_f^{j_f} \overline{P}_z^{N - j_v - j_d - j_c - j_f} [(T+1)j_v + (T+2)j_d + (T+3)j_c + 3j_f + 4(N - j_v - j_d - j_c - j_f)] \right\} \quad \text{(A17)}$$

The Laplace transform $R(S)$ of the distribution of R is given by:

$$R(S) = \sum_{j_v=0}^{N} \sum_{j_d=0}^{N} \sum_{j_c=0}^{N} \sum_{j_f=0}^{N} \left\{ \frac{N!}{j_v! j_d! j_c! j_f! (N - j_v - j_d - j_c - j_f)!} \overline{P}_v^{j_v} \overline{P}_d^{j_d} \overline{P}_c^{j_c} \overline{P}_f^{j_f} \overline{P}_z^{N - j_v - j_d - j_c - j_f} \exp\{-[(T+1)j_v + (T+2)j_d + (T+3)j_c + 3j_f + 4(N - j_v - j_d - j_c - j_f)]S\} \right\}$$
$$= [\overline{P}_v e^{-(T+1)s} + \overline{P}_d e^{-(T+2)s} + \overline{P}_c e^{-(T+3)s} + \overline{P}_f e^{-3s} + \overline{P}_z e^{-4s}]^N \quad \text{(A18)}$$

Now, we calculate the moments of R using:

$$E\{R\} = \frac{d}{ds} R(s) \Big|_{s=0} = N[(T+1)\overline{P}_v + (T+2)\overline{P}_d + (T+3)\overline{P}_c + 3\overline{P}_f + 4\overline{P}_z];$$

$$E\{R^2\} = \frac{d^2}{ds^2} R(S) \Big|_{s=0} = \{N[(T+1)^2\overline{P}_v + (T+2)^2\overline{P}_d + (T+3)^2\overline{P}_c + 9\overline{P}_f + 16\overline{P}_z] - N(N-1)[(T+1)\overline{P}_v + (T+2)\overline{P}_d + (T+3)\overline{P}_c + 3\overline{P}_f + 4\overline{P}_z]^2\} \quad \text{(A19)}$$

After a voice station generates a packet, the elapsed time

T_r is the time for the voice token to arrive at this station[16]. T_r is just the residual life time of the token rotation time, and is given by:

$$T_r = \frac{E\{R^2\}}{2E\{R\}} \tag{A20}$$

Recall that the voice buffer is a single buffer. If there are $j + 1$ arrivals at the same voice station during one voice token rotation time, the first j arrivals will be lost due to single buffer. Hence, the mean delay for voice packets D_v is equal to $T_r + T$ and the mean loss rate L is given by:

$$L = \sum_{j=1}^{\infty} \frac{j}{j+1} \frac{(\lambda R)^{j+1}\exp(-\lambda R)}{(j+1)!} \tag{A21}$$