RC 9157 (#40103) 11/30/81
Communications 51 pages

Research Report

A Feasibility Study of Using Store-and-Forward Data Communication Networks to Transmit Digitized Speech

Dieter Conrads
Kernforschungsanlage Juelich
D-5170 Juelich, P.O. Box 1913
West Germany

Parviz Kermani
IBM, Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

IBM Research Division
San Jose · Yorktown · Zurich

RC 9157 (#40103) 11/30/81
Communications 51 pages

A Feasibility Study of Using Store-and-Forward Data Communication Networks to Transmit Digitized Speech

Dieter Conrads
Kernforschungsanlage Juelich
D-5170 Juelich, P.O. Box 1913
West Germany

Parviz Kermani
IBM, Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

ABSTRACT:
This paper focuses on some issues related to transmission of packetized voice over store-and-forward data communication networks. We begin with discussing the general environment and availability of various hardware, e.g. voice digitizers, needed for this purpose. The issue of error checking of voice packets is discussed next. In contrast to general belief, we show that it is infeasible not to do error checking for voice packets in an environment where voice and data coexist. Furthermore, through mathematical modeling, we demonstrate that there is no significant advantage in providing special, less restrictive, error checking procedures for packetized voice. We then study the capacity and delay requirements in such networks. Through analytical modeling we draw some basic guidelines on how to design a data network to be able to use it for voice transmission as well. It is shown that even with low data rate vocoder type digitizers, high bandwidth channels (i.e. significantly higher than 50 kbps) are required in order to meet the delay requirements of voice communication. However, because of the present day high prices of the required hardware, providing packetized voice transmission is yet an uneconomical venture. In the last part of this report two procedures to provide guaranteed services are outlined. Guaranteed throughput/delay service is not only required for packetized voice in a voice/data network, but is a feature which is desirable to offer in most data networks.

This research was done while Dieter Conrads was on leave at IBM Watson Research Center

ACKNOWLEDGEMENT

The authors would like to thank Dr. K. Bharath-Kumar of IBM T.J. Watson Research Center for his invaluable suggestions and useful discussions.

CONTENTS

1. INTRODUCTION

In the recent past we have witnessed a surge of interest in using store-and-forward[1] data communication networks to transmit digitized voice signals [6]. The basic motivation for the topic is the continuing process of more and more digitizing traditionally analog signals for which voice is no exemption. This general trend has good reasons since digitized voice (and digital signals in general) offer some significant advantages:

- Digital signals in general are less prone to interference (such as cross talk).
- Digitized voice can achieve higher security since it lends itself easily (and may be prerequisite) to encryption/decryption techniques. (In fact, this was the first intention when vocoding techniques were invented.)
- Digital signals can easily and therefore with low cost be handled and manipulated; especially they can directly be processed by digital computers which in case of voice is important for several advanced services.

When voice signals are offered in digitized form there is a chance of handling and transmitting them very much the same way as it is traditionally done with (computer) data traffic. Having a unified approach for handling data and voice traffic could result in significant cost savings. The most important saving most probably would come from the fact of only having one type of network with unified network management, unified hardware techniques (and possibly even components), and unified services. Additional savings could arise from better operational characteristics which mean better utilization of transmission and switching facilities and in general bring about what is called "the economy of scale."

The possibility of handling digitized voice and data in the same way immediately leads to the question which switching technique is optimal for that sort of integrated traffic.

This question has attracted much interest in the recent years but there is still no general accepted consensus. There seem to be some arguments in favor of packet switching. It is ideally suited to handle traffic of bursty nature because it is tailored for that sort of traffic, and should be capable of at least reasonably handling voice traffic because also speech exhibits a bursty character which - appropriate hardware assumed - leads to bursty digitized voice traffic. Indeed there are some papers (e.g. [7], [9]) claiming that for a wide range of parameters (especially for high traffic) packet switching is the optimal scheme. Beyond these arguments we assumed packet switching because we intended to study the impact of adding voice traffic to existing data networks for which packet switching is the usual scheme.

This report contains investigations pertinent to some of the issues that arise from the question whether and under what conditions it is possible to transmit (digitized) voice traffic over existing store and forward data networks, for example, SNA. It turns out that voice traffic has special non-trivial requirements concerning delay and bandwidth and due to its very real time

---

1 In this report we use "store-and-forward switching", "packet switching", and "message switching" synonymously.

nature needs some kind of guaranteed service. So the central problems are how to choose network parameters and how to change network procedures such that the appropriate service can be guaranteed without explicitly reserving resources. Although the question of "guaranteed service" is only discussed in the context of requirements of packetized voice, it is of general importance beyond that special topic. The results presented give a feeling for the interdependencies between voice data rate (VDR), bandwidth, packet length, and delay and can also be used as design criteria.

The second chapter gives an overview over voice digitization techniques and switching techniques and their characteristics. The requirements of digitized voice and the characteristics of the packet switching scheme in general and SNA in specific are discussed. As a first result, some general statements concerning the critical design parameters are derived. In the following chapters some of the issues are discussed in depth. Section 3: "Error Handling" treats alternatives for error handling procedures applying to voice packets in an integrated voice/data network. In Section 4: "Delay Considerations" VDR, packet/packet header length, channel capacity and their interdependencies are studied under the condition that delay is a critical measure. Section 5: "More About Delay" gives a probabilistic approach for studying the maximum queue length and the maximum delay a (voice) packet experiences in a node. In Section 6: "Procedures For Guaranteed Services" two different procedural concepts are presented that guarantee a bounded delay for a complete source/destination connection.

2. DISCUSSION OF THE ENVIRONMENT

The environment we are facing is shown in Figure 1. At the source speech is digitized, it may be encrypted after digitization and is packetized before it is submitted to the store-and-forward network which is also intended to carry non-voice data traffic. At the destination the incoming packets are depacketized and may be delayed in the "adaptive delay buffer" to enforce a more steady flow of information. If applicable, the digital signal is decrypted and then transformed to analog signals and made audible to the listener.

In the following, voice digitization techniques are presented and a description of packet switching scheme and other switching alternatives is given. Finally, the requirements of digitized voice concerning transmission are discussed in the light of the capabilities of existing store-and-forward networks, with some emphasis on SNA.

2.1 VOICE DIGITIZATION

There are two basically different approaches for voice digitization:

1. Waveform reconstruction strategies (quantization).
2. Analysis/synthesis strategies (vocoder/synthesizer technique).

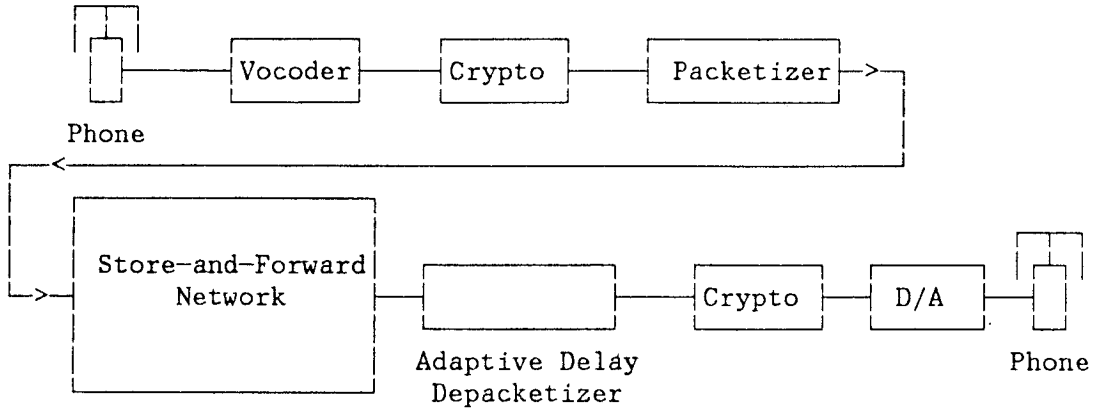

Figure 1. General structure of the environment.

Waveform reconstruction techniques are simple, robust, and cheap (ICs available) but generate a fairly high VDR (voice data rate).

Analysis/synthesis strategies are rather sophisticated and expensive (there are no chips available, but also on chip basis they would be more expensive than quantizers due to the inherent complexity), but produce low VDRs. A number of methods for analyzing (and synthesizing) voice have been proposed and realized; none of them has been proved or accepted to be clearly superior to the others. As a result, until now no standard is evolving.

It is beyond the scope of this paper to describe all the different quantization and vocoding methods in detail, only the basic ideas and characteristics are presented (for detailed description see [3], [8], [10].

2.1.1 Quantization

The basic principle of quantization techniques is to parameterize and later reconstruct the waveform of the original analog signal. The most common quantization method is PCM (pulse-code modulation). The basic scheme is linear PCM, where an A/D converter samples the analog signal (voice) at a constant rate. The sample rate depends on the upper frequency limit of the signal. For telephone quality the voice signal is low pass filtered with 4 kHz. The sampling rate usually is taken twice the limiting frequency (Nyquist frequency) which in this case gives 8 kHz; if the accuracy of the A/D converter is 8 bits, the resulting VDR is 64 kbps.

All the other quantization schemes like log PCM, differential PCM and CVSD (continuously variable slope delta modulation) are derivates of the basic linear PCM scheme. The most popular of these techniques is the 64 kbps log PCM which employs a companding technique in order to reduce the dynamic range of the digitized signal. This scheme is used by the telephone companies (national PTTs) and therefore is easy and cheap to implement.

The most sophisticated - but due to the existence of appropriate ICs now nevertheless cheap to implement - quantization technique is CVSD which typically produces a VDR of 16 kbits.

2.1.2 Analysis/Synthesis Strategies

Analysis/synthesis methods make no attempt to preserve or reconstruct the original waveform. They attempt to parameterize the voice signal according to an underlying model of speech and this information is then used to synthesize a voice which sounds like the original one. All models of speech production assume the separation of the model into an excitation function applied to a vocal-tract transfer function; the latter being viewed as a time-varying linear filter with relatively slow varying parameters (due to speed restrictions of the physical apparatus). The parameters are basically:

Voicing: Voiced or unvoiced.

Pitch: Frequency of the vocal cord vibration.

Gain: Measure of sound intensity.

Spectral envelope: Filter coefficients, describing the vocal tract shape.

The different vocoders (e.g. channel vocoder, linear predictive vocoder (LPS), formant vocoder, cepstrum (homomorphic) vocoder) vary in how they estimate the speech parameters.

Vocoders can easily be combined with silence detection and encryption techniques; they typically produce VDRs between 1.2 and 4.8 kbps.

There has been considerable progress in the development of vocoders in the recent past partly due to technological progress, but they are still very expensive and not developed enough to be commercially applicable.

2.2 SWITCHING ALTERNATIVES

2.2.1 Circuit Switching

In basic circuit switching, a line is switched between the requesting node and the destination; after completion of the process of signalling and reservation the source has exclusive access to that path. The main deficiencies of this scheme come from:

1. The overhead of explicitly switching a line between source and destination.
2. The fact that the switched line is reserved for the source which may not be capable of fully and permanently using the line capacity.

The above considerations indicate that circuit switching is best suited for requirements in which the full line capacity can be used for a nontrivial time interval. This is the case for a constant data rate source (like voice without silence detection) but also for transmission of a large file in batch mode.

Due to physical limitations concerning the number of lines, there cannot be an access guarantee to the network as long as the number of network subscribers exceeds the number of lines provided by the network. If a user gets access to the net, he receives a guaranteed service concerning delay and bandwidth which is derivable from the characteristics of the dedicated line.

There is no provision (from the network equipment) that information entered by the source is truly and error-free delivered to the destination, and a line breakdown interrupts the source/destination connection.

2.2.2 Packet Switching

In a basic packet-switching environment each packet is a self-contained unit that includes all the information the network facilities need in order to forward it correctly from the source to the destination.

The advantage of this scheme is that no network resources are reserved and no switching in a narrow sense is encountered because no path between source and destination is switched. The overhead this scheme experiences comes from the packet header that provides the information that enables the network to handle each packet on an individual basis. The second source of overhead comes from the routing process. In order to be capable of properly (and according to network specifics) moving packets from source to destination, the network has to maintain and dynamically update information about the network topology and its actual status and use it for routing of each individual packet.

From these characteristics it is clear that packet switching is best suited for bursty traffic, especially if the bursts and the interval between them are not too long. On the other hand, unnecessarily high overhead may be encountered for a steady flow of information flowing between a specific source/destination pair.

To date, there has been no notion of any service guarantee concerning delay or throughput (bandwidth) in existing packet-switching networks. There is normally a guarantee of having access to the network and a guarantee for delivering each submitted packet correctly if the specified destination is valid, i.e., if at least one path exists between source and destination. This leads to another advantage of packet switching: A break down of a line should not explicitly affect a user as long as connectivity between source and destination is given.

2.2.3 Hybrid Switching

In most of the existing networks the basic switching schemes described before are more or less modified; the virtual/explicit route concept of SNA, for example, is one step in the direction

to the circuit switching concept. Beyond such modifications there exist two different approaches to really combine the two switching schemes in order to be able to offer the optimal switching technique for any sort of traffic.

The more ambitious approach is commonly referred to as hybrid switching. In this case there exists one network which internally realizes packet switching and circuit switching for an appropriate part of the overall transport capacity. In this scheme (see Figure 2) the time is divided in constant time master frames which again are subdivided into slots, each of which represents a certain fixed amount of transportation capacity. In each frame part of the slots are devoted to circuit switching (e.g. for voice traffic) and part to packet switching. The boundary between the slots allocated to the different switching schemes may or may not be moveable.

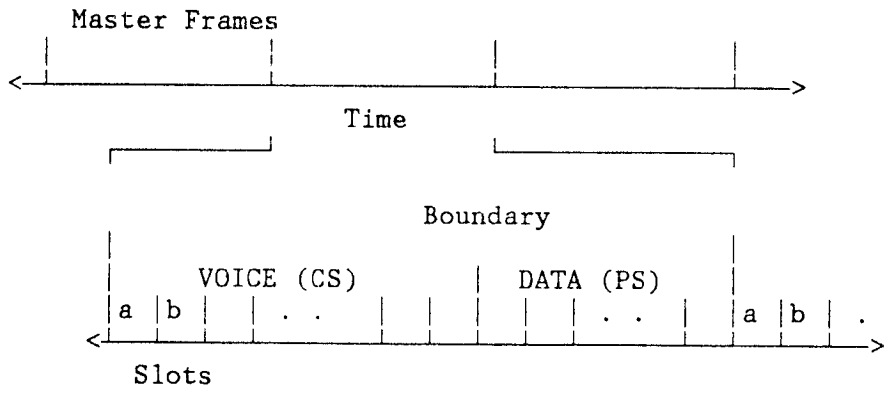

Figure 2. Integrated switching through a master frame scheme.

The basic idea of offering each type of traffic the optimal matching switching scheme within one single network is very appealing but the approach has some critical points:

- If a fixed boundary strategy is used, it may not match the dynamically changing traffic requirements thereby losing a lot of the potential benefits of the approach.
- If a movable boundary strategy is employed, the task of finding an algorithm that controls the movement such that it really reflects the need of the changing traffic pattern is by far nontrivial.
- The implementation cost of that approach is high in terms of hardware and software leading to comparatively high-switching costs. The trade-off between high-switching costs and the potential gain may not justify the effort.

The second approach for offering distinct switching capabilities might be called alternative switching. In this approach (see Figure 3) two separate networks for packet switching and circuit switching are maintained. Added to the environment is a switching facility that (besides local switching) is capable of dynamically switching each incoming line to either

network depending on the actual application. From a technical point of view this approach cannot be a final solution, because it not only maintains two different kinds of networks, but furthermore adds another nontrivial device. On the other hand, it may be a useful evolutionary step because it needs no changes in existing environment but nevertheless from the point of view of a user it represents a remarkable progress because it looks like a unified approach.

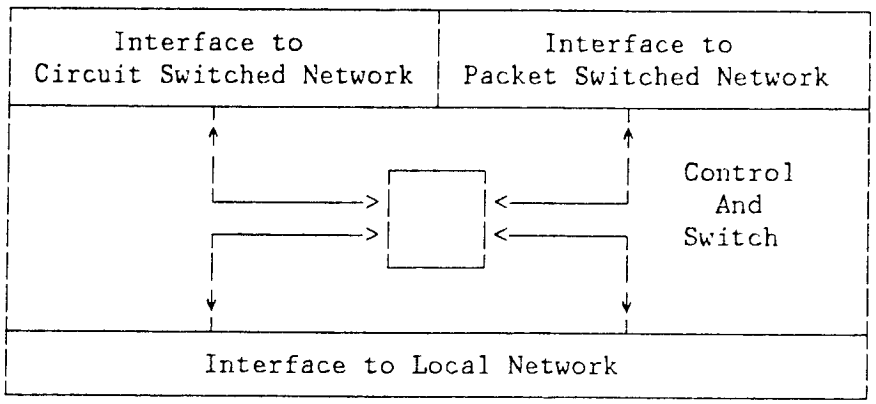

Figure 3. Alternative switching.

2.3 ISSUE OF PACKETIZED VOICE TRANSMISSION

In using store-and-forward networks for packetized voice transmission, one has to consider some issues and certain requirements. In the remainder of this section we briefly elaborate on these issues and requirements in the light of capabilities of existing store-and-forward networks like SNA. In later sections of this report we study these items in more depth and more quantitatively.

As pointed out earlier, one of the basic characteristics of a store-and-forward network is its capability to deliver packets to their destinations error free. This is done through different error handling schemes. These schemes generally introduce some overhead and using them invariably increases the network delay. It has been argued that error handling can be deemphasized in the case of packetized voice, the claim being minor errors only slightly affect the quality of voice [4]. This issue is the subject of our study in Section 3: "Error Handling", where we show that within a data communication network environment there is no substantial gain in changing error handling procedures specifically for packetized voice.

The basic requirements of digitized voice concerning transmission can be deduced from the characteristics of speech and the digitization process:

- Non trivial to high bandwidth requirements.
- Steady flow of information and the issue of service guarantee.
- Low overall transmission delay (1/2 seconds is regarded to be acceptable for a telephone conversation)

Bandwidth The maximum bandwidth per line in existing networks is usually around 50 kbps. As long as there is not a major improvement of this value, there is no chance of using the most developed and popular high VDR digitization schemes within such networks. So the only reasonable possibility for adding voice traffic is adopting vocoder techniques for voice digitization. The disadvantage - as was pointed out in 2.1.2: "Analysis/Synthesis Strategies" - is that these techniques are not yet developed to commercial applicability. In addition, employing a low VDR source in conjunction with low capacity links may create severe problems concerning delay as discussed in Delay.

Steady Flow of Information As was mentioned in 2.2.2: "Packet Switching", up to now there is nothing implemented in existing packet-switched networks that guarantees a certain level of service whether the criterion is bandwidth (throughput) or delay or flow characteristics. Moreover, pacing mechanisms, as a mean of network flow control, conflict directly with the issue of steady flow. In fact, the basic assumption when employing pacing mechanisms is that the data source can adapt (restrict) itself to the limitations of the destination (in order to avoid overload conditions) and/or to the actual load conditions of the network (in order to avoid network congestion). This assumption, in general, is no longer valid for digitized voice input. The topic of flow control and service guarantee is of extreme importance in this environment. This issue will be addressed in later sections of this report.

Delay For three reasons delay is a major problem in networks with characteristics like SNA:

1. Packetization delay is increasing with decreasing VDR.
2. Transmission delay is increasing with decreasing bandwidth and the number of hops.
3. Queuing delay is increasing with decreasing bandwidth.

Reducing the packet length reduces delay due to all of the above reasons. On the other hand - as each packet needs a packet header which finally is overhead - there is a trade-off between packet length and efficiency. Especially in SNA with its very large message header (26 bytes) efficiency may very soon become a critical point when reducing packet length. In general, also the computational overhead in each node may no longer be neglected (which is normally done) if more and more packets have to be processed in order to transport a certain amount of information.

The above discussion shows that the capabilities of *existing* packet-switching networks do not match the requirements of digitized voice too well. Besides the fact that some of the existing network procedures (like those for flow control) need to be revised, delay is the critical point when using low VDR digitization devices in a low bandwidth network environment and the interdependencies between delay, packet length, VDR, and channel capacity need to be studied carefully. In the coming sections of this report we will evaluate these issues and study

the necessary changes to existing store-and-forward networks to be able to use them for packetized voice transmission.

3. ERROR HANDLING

In existing data networks each bit submitted to the network is regarded to be valuable; therefore, every detected error causes a retransmission of the packet in error. In the case of voice packets one can argue that an error, or even the loss of a packet- if the amount of real time speech included in one packet is not too high -will not significantly reduce intelligibility. Therefore, in this case one could ignore an error or discard an erroneous packet in order to avoid the additional delay due to retransmission and improve transmission efficiency [4]. Unfortunately such a scheme is not easy to implement in an environment intended to carry mixed voice and data traffic.

Error checking normally is realized by a CRC which the sender calculates and appends to the information to be CRC protected. The receiver does the same calculation and compares its results with that received from the transmitter; a discrepancy indicates an error.

The main point concerning the considerations above is that if the receiver detects an error, it cannot trust the received information and thus is not capable to find out whether the received erroneous packet is a voice packet, in which case it simply could discard it or send it towards the destination without requesting for a correct copy, or a data packet, in which case it has to call for retransmission after discarding it.

There are two possible ways to ignore error and/or discard erroneous voice packets:

1. By provision of mechanisms provided by the sending node.
2. By introduction of a second CRC for the header.

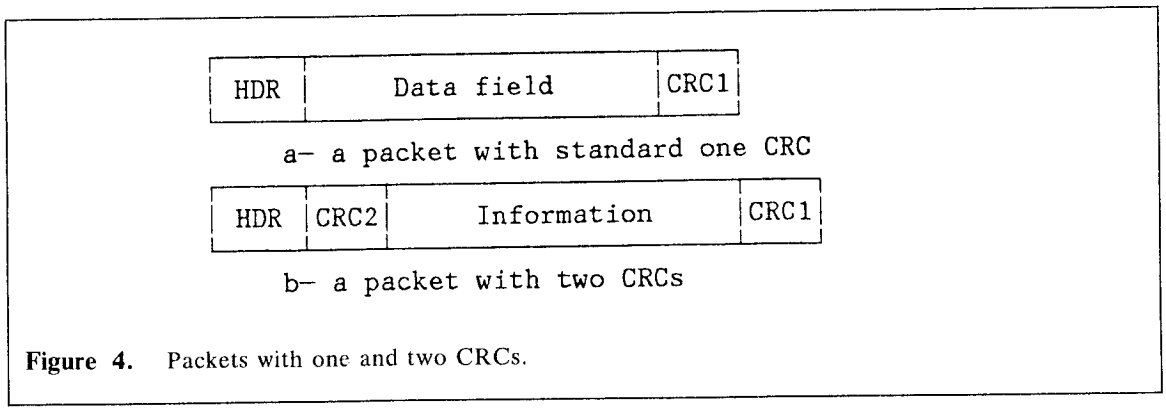

Figure 4. Packets with one and two CRCs.

In the first alternative the receiving node signals back the detection of an error. Upon the receipt of the error message the transmitting node makes the decision whether to ignore the

error message (in case of voice packets) or to retransmit the packet (in case of data packets). There are some drawbacks inherent with this solution. When a transmitting node decides not to retransmit a packet it may have to reassign sequence numbers to all the remaining packets (these sequence numbers are used at the link level in protocols such as SDLC). The reassignment of sequence numbers may prove to be a nontrivial task and can introduce extra overhead and delay. The second draw back stems from the fact that in this scheme basically only error free packets are transmitted out of a node. It may well happen that there are several consecutive voice packets with error and dropping them will result in a large real time gap in voice packet stream and consequently cause a drastic deterioration of the voice quality. Finally, if the supporting network is architected such that the sequence of packets is preserved at every intermediate node (as in SNA), this scheme does not reduce the delay by much.

With the above considerations we do not find this alternative a plausible approach and hence elaborate on the second alternative throughout the rest of this section.

Introduction of a second CRC for the packet header gives the possibility to check the correctness of the control information (header) even if the whole packet is erroneous (Figure 4). The rational behind this being the fact that in general the header is much smaller than the data part of a packet.

If the header is in error (this is detected through CRC2, Figure 4), the complete packet is requested for retransmission. This is similar to standard error handling, but there is a non-negligible gain in delay because the detection takes place right after CRC2 is received.

If an error occurs in the data field and the header is error free the receiving node can make the decision as to discarding the packet or requesting for retransmission depending on the type of packet. This may significantly reduce the delay for voice traffic. Furthermore, it may be argued that because unnecessary retransmissions are avoided, the transmission efficiency is also improved. By this approach it would also be possible to forward the erroneous packet.

As the above discussion shows the introduction of an additional CRC for the packet header may lead to a reduced delay and improved transmission efficiency. On the other hand this implies a significant change in most of the existing link protocols (such as SDLC, HDLC, ADCCP) and the supporting hardware or firmware. Also for error free packets the additional overhead per packet because of the second CRC is 16 bits; hence the transmission efficiency is degraded.

To study the gain in transmission efficiency with the 2__CRC scheme and compare it with the standard 1__CRC scheme an analytic model was developed; details of this model are presented in Appendix A: "Performance of Two Error Checking Schemes.".

Intuitively, on a highly noisy link, the 2__CRC scheme saves a lot of retransmission of voice packets, so one might expect a higher transmission efficiency when voice traffic is high. Figure 5 shows the ratio of efficiency of transmission of the two schemes for different error rates as a function of relative volume of voice traffic. It is noticed that for an extremely high error rate ($P_e = 10^{-3}$) the 2__CRC scheme is much superior to the standard scheme; however, this error rate is unrealistic. For less noisy links ($P_e = 10^{-4}$) the gain is marginal, and finally for an error rate which is usually experienced in medium quality links

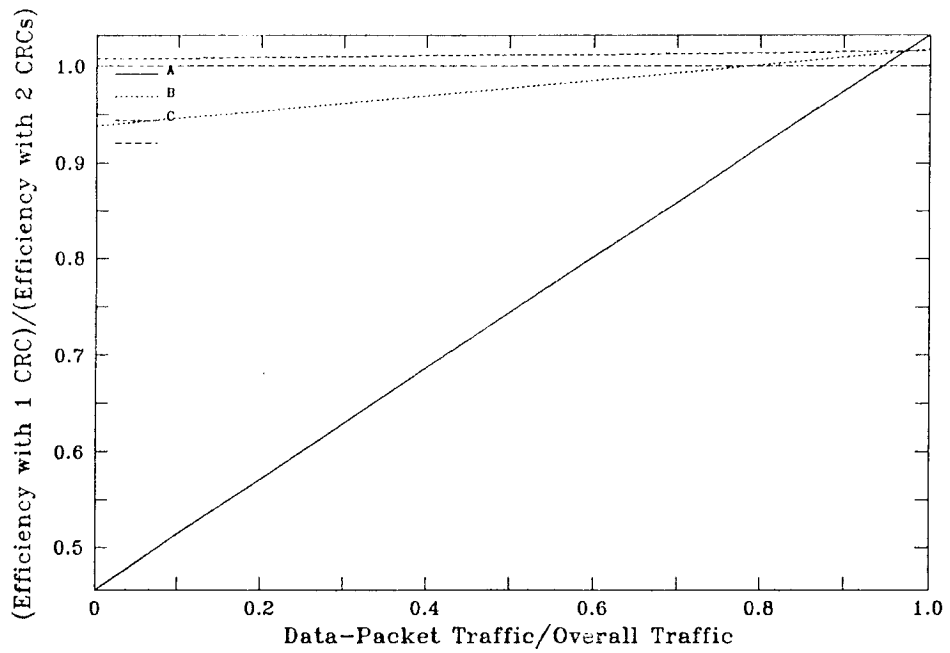

Figure 5. Relative efficiency of two error handling schemes for varying error rates : Packet length=100 Bytes, header length=26 bytes, CRC1=CRC2=16 bits, A: $P_e = 10^{-3}$; B: $P_e = 10^{-4}$; C: $P_e = 10^{-5}$; D: Line of equal efficiencies.

($P_e = 10^{-5}$), there is no gain in using the 2__CRC scheme. Therefore, the overhead for the second CRC is only justified for links with very high error rates and even then the gain is marginal. (The efficiency argument is not applicable to protocols like DDCMP, used in DECnet, that already employ a separate CRC for the header for other purposes.) This point will become even more critical in the future as we expect an increased usage of high quality (low error rate) lines (coax and fiber optics).

Figure 6 shows the effect of length of data field of a packet on the efficiency of transmission. The error rate for all the cases studied in this figure is $P_e = 10^{-4}$ which is higher than normal. This figure shows that the 2__CRC scheme gives better performance for larger packet length; however, the gain is very insignificant. For more realistic error rates, the standard 1__CRC scheme always results in a better performance.

Figure 7 shows the effect of header length on the performance. All observations as in Figure 6 apply to this case as well.

Because the header is usually substantially shorter than the data part of a packet, one may argue in favor of providing for a smaller number of CRC bits for the header. In Figure 8 we show the relative efficiency of the two error handling schemes for different CRC lengths for the header. This figure, as before, shows that the gain is marginal. One should note that using a CRC of different length requires provision of non-standard hardware and/or

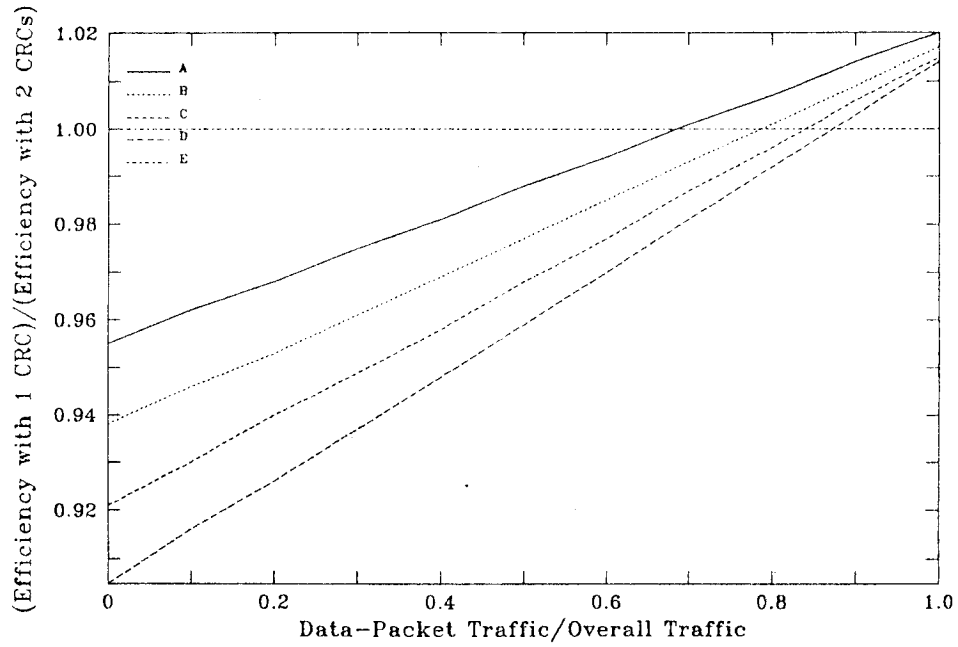

Figure 6. Relative efficiency of two error handling schemes for varying packet lengths : Header length = 26 bytes, CRC1 = CRC2 = 16 bits, $P_e = 10^{-4}$, A: packet length = 80 bytes; B: packet length = 100 bytes; C: packet length = 120 bytes; D: packet length = 140 bytes; E: Line of equal efficiencies.

additional programming which, from cost point of view, offsets the marginal advantage that the smaller header length may have.

Beyond the aspects already discussed, there are more general points which should be taken into account:

- Implementing two different error handling schemes in each node of a network is per se a non trivial overhead. Furthermore, when employing a layered network architecture it violates the basic idea of this principle because the lower network layers (in this case the link level) need to have some knowledge about the contents of the packets they are forwarding.
- If a network makes use of the fact that losing a voice packet does not significantly reduce the intelligibility, then it should take care that no two, or more, adjacent or near neighbor packets of the same voice channel can be lost. The probability of two adjacent packets of the same voice channel being in error is not very high, but may not be negligible either, because errors tend to have a bursty nature. Guaranteeing that no two adjacent or near neighbor packets be dropped is a nontrivial task and under certain assumptions completely impossible.
- If it is really true that losing packets up to certain percentage does not significantly degrade intelligibility of speech, that indicates that more information than necessary is

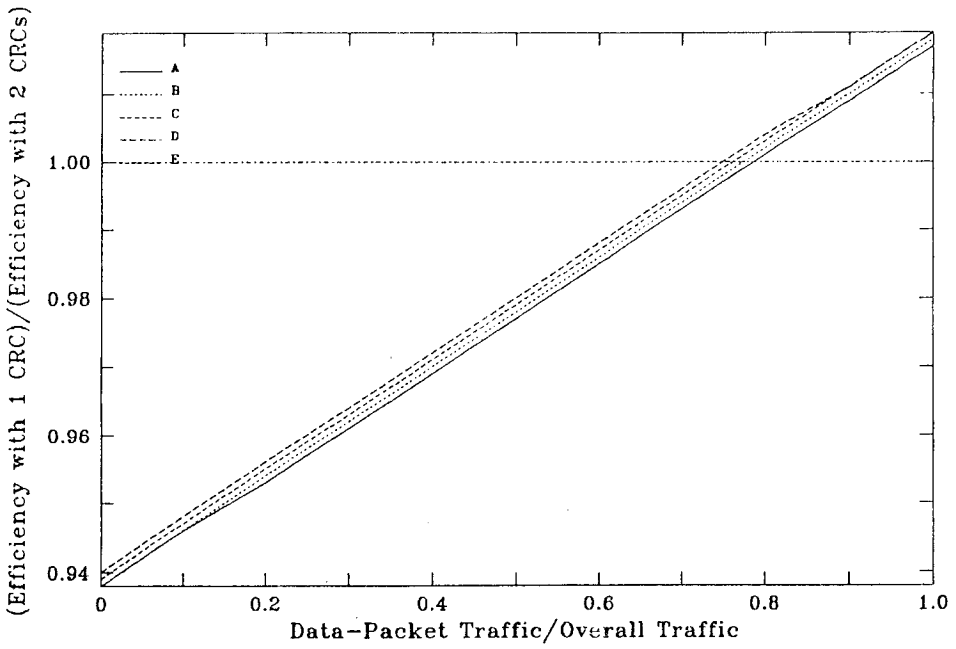

Figure 7. Relative efficiency of two error handling schemes for varying header lengths : Packet length = 100 bytes, CRC1 = CRC2 = 16 bits, $P_e = 10^{-4}$, A: header length = 26 bytes; B: header length = 20 bytes; C: header length = 16 bytes; D: header length = 10 bytes; E: Line of equal efficiencies.

generated and transmitted. However, it would then make sense to remove this redundancy in favor of more efficient transmission.

These considerations become important in the light of recent developments in voice digitization techniques which have brought about low VDRs. Generally, by decreasing the VDR the amount of real time speech contained in one packet is increased and obviously there exists a limit when loss of a packet is no longer acceptable. In addition to this general aspect, some of the common techniques to reduce VDR lead to interdependencies between several packets, hence losing one packet can influence a fairly great amount of real time speech which is far beyond that contained in that one packet. Examples for these techniques are: sending the value of a parameter only if it has changed beyond a certain threshold; using some sort of delta information instead of absolute values; or employing talk-spurt/silence detection which may result in appropriate indicator bits. All these indicate that the ultimate assumption that losing a single packet does not seriously degrade intelligibility need to be carefully proved when using low VDR digitization techniques.

In the light of the discussions of this section, the final conclusion is that providing a separate error handling procedure for voice packets

- does not necessarily improve the transmission efficiency by much,

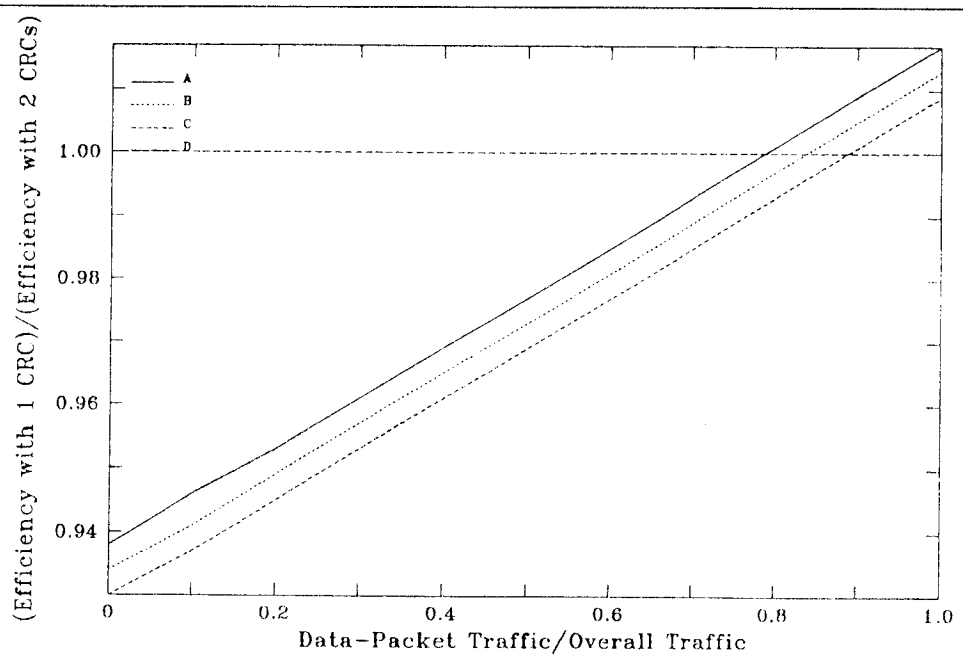

Figure 8. Relative efficiency of two error handling schemes for varying CRC2 lengths : Packet length = 100 bytes, header length = 26 bytes, $P_e = 10^{-4}$, A: CRC2 length = 16 bits; B: CRC2 length = 12 bits; C: CRC2 length = 8 bits; C: Line of equal efficiencies.

- requires special hardware which may prove costly, and
- in a low speed VDR environment may seriously degrades the quality of voice.

As such, a standard error handling scheme is a right choice for data as well as packetized voice traffic.

4. DELAY CONSIDERATIONS

In this section we study the design parameters which impact the delay which voice packets experience in passing through a network. We first elaborate on the design parameters in Section 4.1: "Basic Issues" and introduce a simple analytic model which we use for our study. In Section 4.2: "Numerical Examples" we then present some numerical examples based on this model, and finally in Section 4.3: "Some Conclusions" we draw some conclusions based on these examples.

4.1 BASIC ISSUES

The network delay is affected by different parameters of the system, such as VDR, voice packet length, presence and volume of interfering data traffic, network topology (more specifically the number of hops a voice packet travels), network channel bandwidth, etc... The main purpose of this section is to study the interdependencies of these parameters.

Figure 1 in Section 2: "Discussion of the Environment" shows the general structure of the environment under study. As we pointed out there, in order to to provide the proper grade of service to voice traffic, this class of traffic is given higher priority over data traffic.

The major sources of delay in such an environment are

1. Digitization and packetization delay.
2. Queueing and processing delay at intermediate nodes.
3. Transmission delay.
4. Propagation delay.

Analog voice signals are first converted to a digital bit stream. The bit stream is then assembled into packets of some maximum size and delivered to the network for transmission to the destination. The rate of digitization (VDR) and the packet size have direct effect on the delay. Basically a high VDR results in less delay, because it takes less time to assemble a packet. On the other hand a high VDR is not desirable because it consumes a large portion of channel capacity.

Considering the packet size, for a large packet size it takes a long time to assemble digitized voice into packets. Furthermore, as we pointed out earlier, if a voice packet encounters a longer delay than the maximum allowed, it is dropped and is not delivered to the receiver. If the packet size is large, dropping a packet results in creating a large gap in voice which may seriously degrade the voice quality. Finally, a large packet size requires a large amount of buffer storage in the "adaptive delay buffer" at the destination node.

Albeit these considerations, one should not perceive that the packet size should be as small as possible. A small packet size results in low transmission efficiency (each packet carries with it a certain amount of control and addressing information); moreover, it results in generation of a large number of packets for a talk spurt. This creates a high arrival rate of packets the processing of which may exceed the computational power of the nodes. A simple calculation reveals that with the present day technology intermediate nodes do not have the power to handle this load.

The nodal and transmission delay also constitute a large portion of the overall delay. Nodal delay consists of queueing for the processor, processing, and queueing for transmission. Our assumption is that voice traffic has priority over data traffic. Also, assuming the nodal processing power is high enough to handle the traffic, the queueing delay for the transmission channels becomes the major source of delay. One can also reduce the transmission queueing delay by using high capacity links (which of course reduces the transmission delay as well).

However, there is a limit on how large a bandwidth one can use. The main bottleneck is, again, the nodal processing power. With high bandwidth the network channels become capable of transmitting a large volume of traffic; however, with present day technology nodes usually cannot handle this amount of traffic.

Propagation delay becomes important when the network encompasses a vast geographic area and uses long haul links. This delay is the result of the finite propagation speed of electromagnetic signals. Because the nature of this delay is not a result of the system design, and is constant for a given terrestrial link irrespective of other parameters, we will not incorporate this delay in our study. We should, however, point out that propagation delay becomes critical and introduces major constraints for satellite links.

To study the effects and interdependencies of these system parameters, a simple analytic model is developed. An analytic model which incorporates all these parameters is, by nature, a complex one; as such, the environment is amenable to analysis only after accepting a fair amount of simplifying assumptions and abstractions. Some of these assumptions, in the first glance, may seem quite unrealistic; however, the resulting model will prove to be a useful tool to study the gross impact of the interdependencies of these parameters. Furthermore, the model can be used to derive some general and useful conclusions. Details of this model are presented in Appendix B: "Maximum Delay Under Deterministic Assumptions". The model is used to study the *maximum* delay when a certain number of speech sources are *continuously* active. Therefore, these sources are constantly generating a bit stream of data, the rate of which is determined by the VDR. In the model all voice traffic travel over the same number of tandem nodes (number of hops), see Figure B.1. It is assumed that the load on all nodes on the path is the same and that the capacities of the links between adjacent nodes are the identical. The maximum delay at a node is encountered when all voice channels are active and continuously generating traffic. In the worse case the delay of a voice packet is the sum of the transmission time of of a data packet (the priority is non preemptive) and of one voice packet per every other sources.

In the discussions below, we use the following notations:

V voice data rate, VDR
C :channel capacity (between nodes)
β : fraction of channel capacity to be used by voice
D : maximum delay
l_v : voice packet length
l_d : data packet length
l_h : header length (for voice and data)
n_h : number of hops
n_v : number of voice channels

The end-to-end delay in our model has the following components:

1. Packetization delay: This delay depends on VDR, V, and voice packet length, l_v.

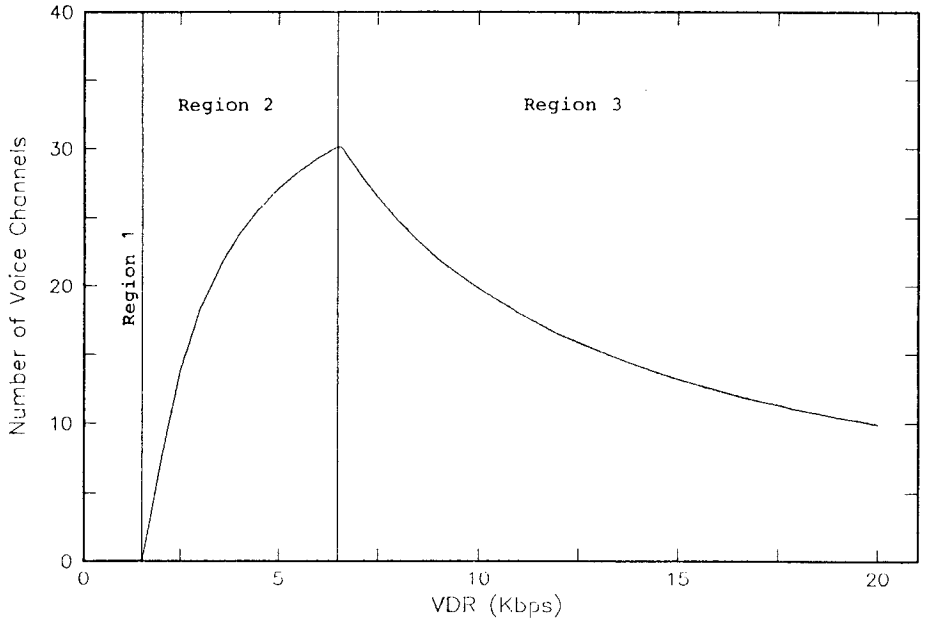

Figure 9. General Shape of number of voice channels vs. VDR curve : parameters of this curve are $l_v = l_d = 800$ bits, $l_h = 26$ bytes, $D_{max} = .5$ seconds, $C = 250$ kbps, $\beta = 1$.

2. Transmission delay: This delay depends on the channel capacity, voice packet length, header length and the path length (C, l_v ,l_h, and n_h, respectively).

3. Queueing delay due to voice traffic: This delay depends on C, l_v, l_h, n_h, and n_v, number of voice channels.

4. Interference delay: This delay depends on the data packet lengths, l_d, the header length l_h, the channel capacity C, and the path length n_h.

The dependency of the maximum delay D on these components is derived in Appendix B: "Maximum Delay Under Deterministic Assumptions" to be:

$$D = \frac{l_v}{V} + [\frac{l_d + l_h}{C} + n_v\frac{l_v + l_h}{C}]n_h \tag{1}$$

With the following constraint:

$$n_v V \frac{l_v + l_h}{l_v} \leq \beta C \tag{2}$$

One way to make use of these results is to establish an upper bound on voice packet delay, D_{max}, and choose the other parameters (n_v, V, C, etc.) to realize this bound. To study the interplay of the design parameters, we show the number of voice channels which can be supported as a function of V, the VDR, for a given D_{max}, when other parameters are kept constant.

The general shape of such a curve is shown in Figure 9 which consists of three adjacent regions. These regions are:

Region 1: When the VDR is too low, packetization delay is excessive and no voice channel can be supported to realize the D_{max} delay.

Region 2: In this section the end-to-end delay is the limiting factor on the number of voice channels. This section is the most important one in which all parameters interact with each other. Region 2 is surrounded on the two sides by regions 1 and 3.

Region 3: In this region the VDR is high enough so that the transportation delay determined by Eq. (1) is well within the desired range (i.e. $D \leq D_{max}$); however, the number of voice channels is restricted by the bandwidth available, which is determined by Eq. (2). The general shape of the curve in this range follows a hyperbolic function ($1/V$, with V varying).[2]

4.2 NUMERICAL EXAMPLES

To start with we choose $D_{max} = .5$ seconds. It has been suggested that .5 seconds is the maximum delay a voice packet can tolerate before the quality of the transmitted voice conversation becomes severely degraded. In Figure 10 we show the number of voice channels and the fraction of capacity actually used by voice traffic as a function of VDR for different channel capacities, C. Other parameters are specified in the figure. When delay is the limiting factor (region 2), the number of voice channels increases faster than channel capacity. For example increasing C from 50 to 100 kbps, when V = 5 kbps, more than doubles the number of voice channels. The same observation is true for the portion of capacity used by voice. However, when capacity is the bottleneck (region 3), the number of voice channels increases linearly with C. Also note that the VDR which maximizes n_v is smaller for larger capacities. This fact is important in designing networks with small channel capacity, as in this case to achieve maximum N_v one has to use relatively high VDR; however, for low C, one cannot afford to use high VDR.

The effect of voice packet length, l_v is shown in Figure 11. The environment in this figure is basically the same as the one in Figure 10, but the channel capacity, C, is set to 50 kbps, except for curve D, for which C=100 kbps. Note that in all cases $l_v = l_d$. This figure clearly

---

[2] Because in region 2 the curve is monotonically increasing with VDR and in region 3 it is monotonically decreasing, the maximum number of voice channels is always reached for the VDR at which region 2 and 3 meet.

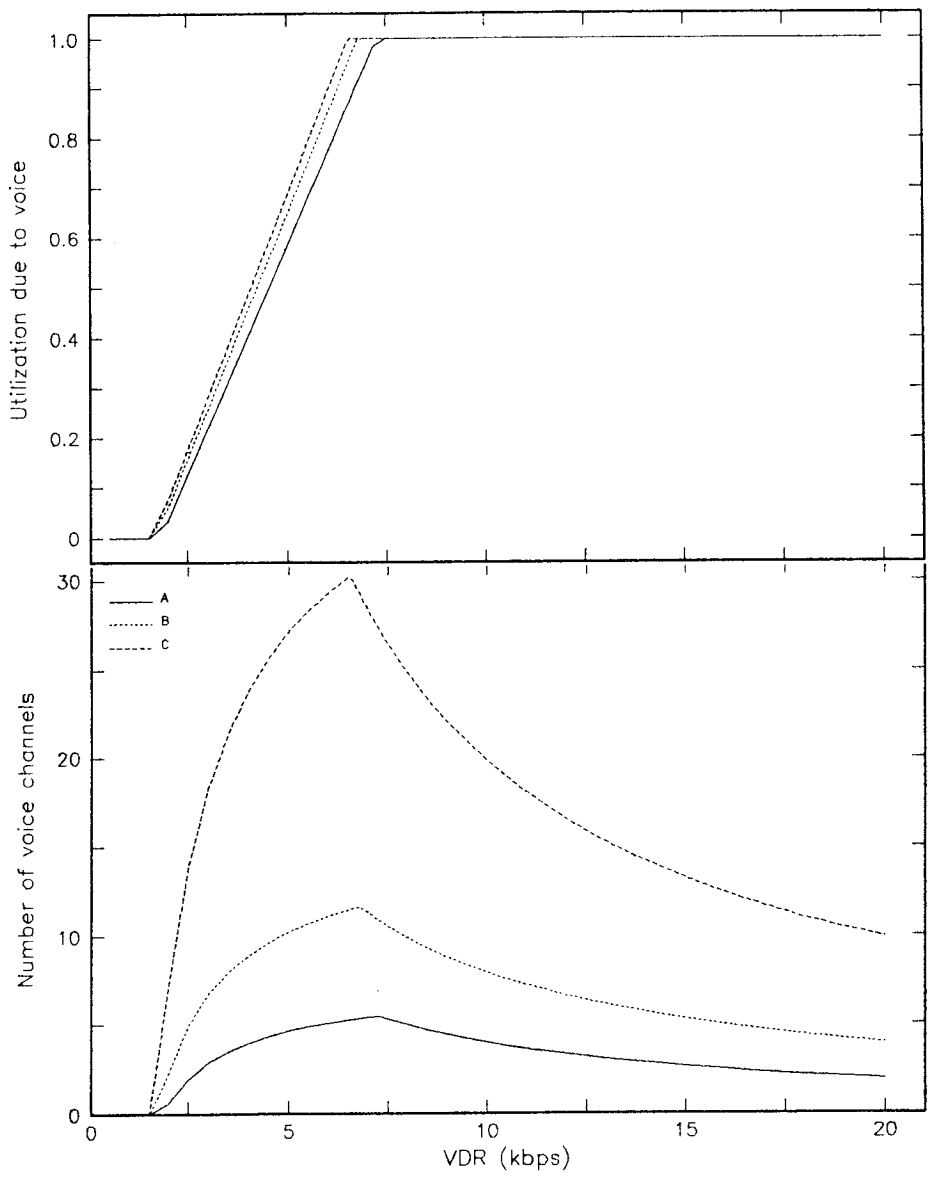

Figure 10. Number of voice channels and utilization due to voice traffic vs. VDR for given capacities : $l_v = l_d =$ 800 bits, l_h = 26 bytes, $D_{\max}$ = 0.5 seconds, β=1, A:C = 50 kbps; B:C = 100 kbps; C:C = 250 kbps.

shows the significance of voice packet length in the system design. Understandably, when

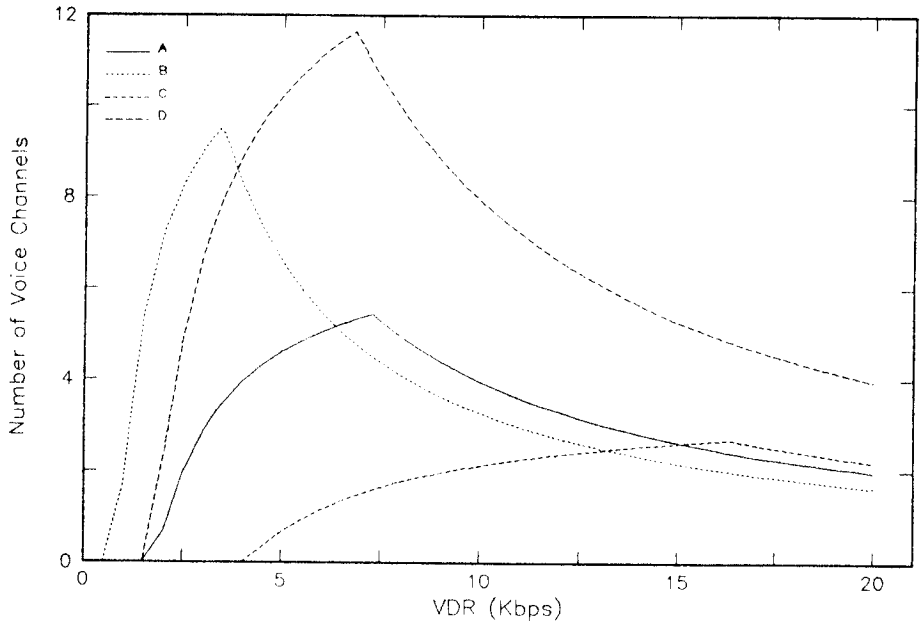

Figure 11. Number of voice channels vs. VDR for different packet lengths : l_h = 26 bytes, D_{max} = 0.5 seconds, β=1, A: l_v = l_d = 800 bits, C = 50 kbps, B: l_v = l_d = 400 bits, C = 50 kbps, C: l_v = l_d = 1600 bits, C = 50 kbps, D: l_v = l_d = 800 bits, C = 100 kbps.

delay is the limiting factor (regions 1 and 2), smaller l_v results in larger n_v. However, this trend is reversed when capacity is the bottleneck. The reason for this phenomenon is that for larger voice packet length, l_v, the transmission efficiency is higher (see Eq. (2))

The effect of the number of hops, n_h, on n_v is shown in Figure 12 and Figure 13 for C = 50 kbps and C = 250 kbps, respectively. The packetization delay is invariant of the number of hops; however, the other components of delay are linearly dependent on n_h. As a result, the curve which specifies the region where capacity is the bottleneck (region 3) is the same for all cases. However, the delay region (region 2) is different between these cases. The reason is, when n_h increases, the amount of delay a packet is allowed to undergo in each node is decreased. In fact the allowed nodal delay is inversely proportional to n_h. These figures show that the general shape of the curves is very much the same for C = 50 kbps and C = 250 kbps except that the maximum of the curves (that is the lowest VDR for which channel utilization is 100 percent) for the lower channel capacity is more shifted to the right with increasing number of hops for C=50 kbps than the respective points of the curves for C=250 kbps. This is the effect of interference by data packets. Both figures show that the number of hops has a significant influence on the number of voice channels that can be supported in a given environment.

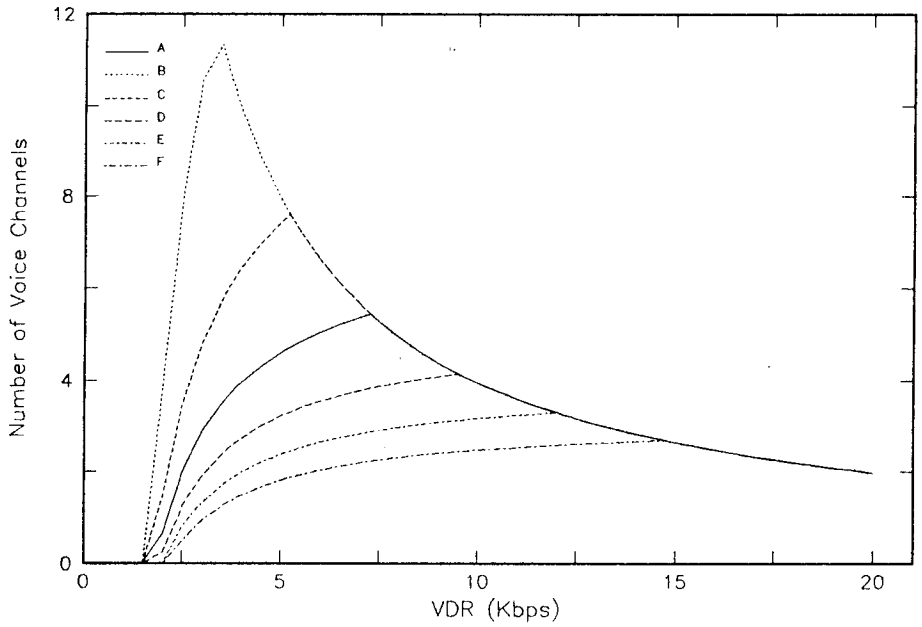

Figure 12. Number of voice channels vs. VDR for different number of hops : $l_v = l_d =$ 800 bits, $l_h =$ 26 bytes, $D_{max} =$ 0.5 seconds, $\beta = 1$, $C = 50$ kbps, A: $n_h = 3$, B: $n_h = 1$, C: $n_h = 2$, D: $n_h = 4$, E: $n_h = 5$, F: $n_h = 6$.

In order to study the influence of the header length the curves for $l_h = 26$ bytes (normal case) are compared with those for $l_h = 8$ bytes which can be regarded to be a fairly small header in a general purpose network. This comparison was made for voice packet lengths of $l_v = l_d = 400$ bits and $l_v = l_d = 800$ bits and for channel capacities of $C = 50$ kbps (Figure 14) and $C = 250$ kbps (Figure 15). As one would expect, reducing the header length from 26 to 8 bytes for $l_v = 400$ bits increases the number of voice in the relative as well as absolute sense. This is a natural consequence of the fact that the reduction of l_h reduces the overhead from 34.2% to 13.8% for $l_v = 400$ bits instead of from 20.6% to 7.4% for $l_v = 800$ bits. Changes in the header length only influence the transportation related types of delay, the packetization delay is not affected.

All in all, the influence of the header length is moderate. Of course, reduction of the header length increases efficiency which can be seen when comparing the right parts of the curves where channel capacity is the limiting factor. It can also be seen from Figure 14 and Figure 15 that there are no specific differences for different channel capacities.

In Figure 16 the dependency of n_v on β is studied. Three channel capacities are considered and in each case β is specified such that the bandwidth available to voice traffic (βC) is 50 kbps. This makes the curves identical for the range of of VDR where bandwidth is the limiting factor (in fact this portion of the curve is identical to the curve of C=50 kbps in Figure 10); however, when delay is the critical factor, the curves show a completely different

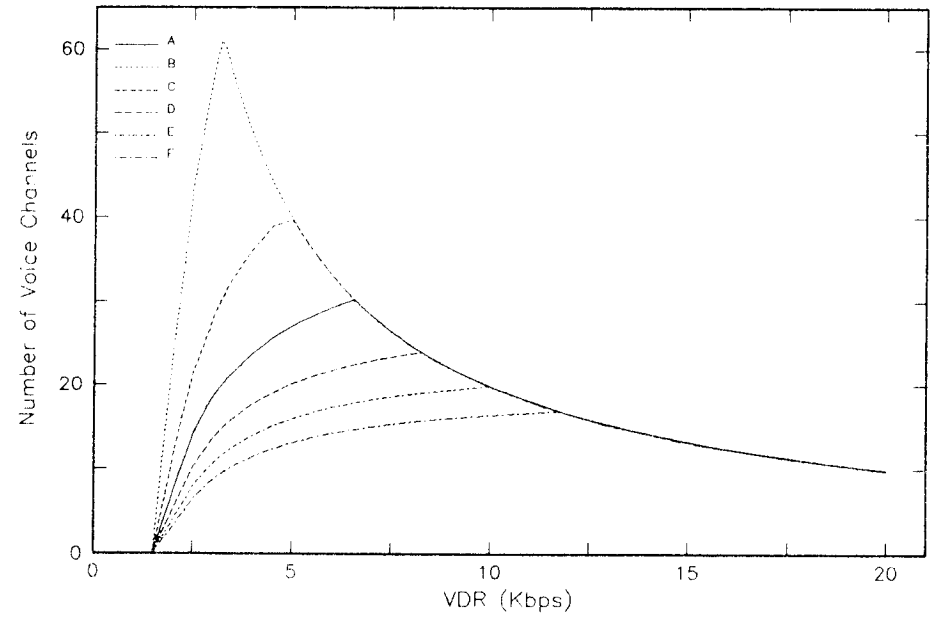

Figure 13. Number of voice channels vs. VDR for different number of hops : $l_v = l_d = 800$ bits, $l_h = 26$ bytes, $D_{max} = .5$ seconds $\beta = 1$, $C = 250$ kbps, A: $n_h = 3$, B: $n_h = 1$, C: $n_h = 2$, D: $n_h = 4$, E: $n_h = 5$, F: $n_h = 6$.

behavior. Actually the curve for C=250 kbps in Figure 16 is identical to the curve for C = 250 kbps in Figure 10 until the bandwidth limitation of 50 kbps becomes effective; the same statement is true for the 100 kbps curve. These curves show clearly that it is advantageous to have integrated services where only part of a higher bandwidth is devoted to voice, and this advantage becomes extremely important in an environment where delay is a critical factor.

We now investigate the effect of varying D_{max} on n_v (see Figure 17). Because packet length (l_v and/or l_d) influence the delay, one might expect some similarities between the curves for n_v for varying D_{max} and varying l_v as shown in Figure 11. In fact any dissimilarity is because of the fact that we have kept the header length (l_h) constant in the latter figure. When doubling the maximum delay we see a curve which is superior to that when dividing the packet length by two (l_v) because no additional overhead is introduced by the former measure; and dividing the maximum delay by two results in a curve which is inferior to that of doubling voice packet length because there is no gain in efficiency introduced by this measure. As a matter of fact, the differences between the curve pairs exactly reflect the loss and/or gain in efficiency when changing packet lengths. As the variations of the maximum delay are independent of the line capacity, the curves for different D_{max} are identical for VDRs where channel capacity is the limiting factor.

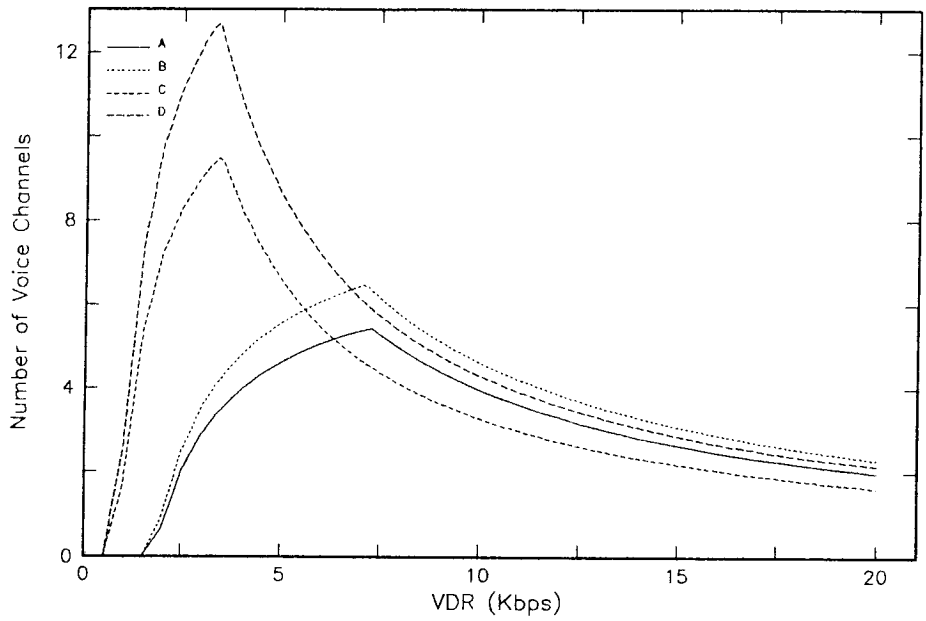

Figure 14. Number of voice channels vs. VDR for different header lengths : $D_{max} = 0.5$ seconds, β=1. C=50 kbps, $n_h = 3$, A: $l_v = l_d = 800$ bits, $l_h = 26$ bytes; B: $l_v = l_d = 800$ bits, $l_h = 8$ bytes; C: $l_v = l_d = 400$ bits, $l_h = 26$ bytes; D: $l_v = l_d = 400$ bits, $l_h = 8$ bytes.

4.3 SOME CONCLUSIONS

The model we used in our study provided us with a general view of the performance of the system. The parameters in our numerical examples were chosen to be as realistic as possible. For example, we did not experiment with channel capacities of mega bits per second order; these channel capacities, although commercially available, can not be handled with the present day hardware technology for switching nodes.

A review of the performance graphs reveal the fact that the maximum number of voice channels, n_v, that the system can support is very small. For example, Figure 15 shows that n_v can be at most 65. One should note that to obtain this number a capacity of 250 kbps is required; this is a capacity that not too many systems can support. Furthermore, to achieve a large n_v one should incorporate digitizers (vocoders) with data rates much lower that the standard 64 kbps of PCM (which is commonly used in telephony). Considering the status of the present day technology in voice digitization hardware, we find out that devices in the VDR range required are very expensive.

All in all, the findings in this section seem to recommend that with present day technology, transportation of a large volume of packetized voice through store-and-forward data networks, is physically infeasible and commercially impractical.

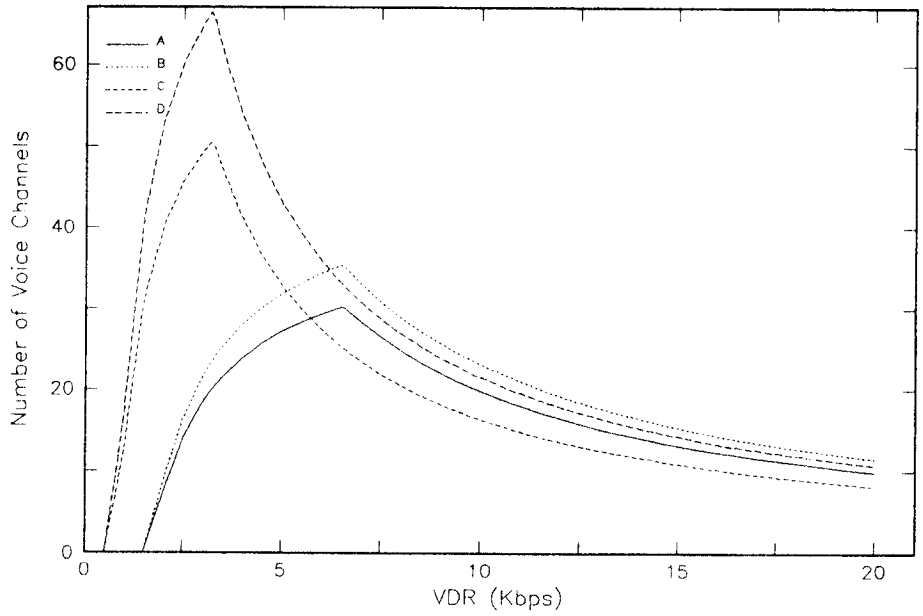

Figure 15. Number of voice channels vs. VDR for different header lengths : $D_{max} = .5$ seconds $\beta = 1$, $C = 250$ kbps, $n_h = 3$, A: $l_v = l_d = 800$ bits, $l_h = 26$ bytes; B: $l_v = l_d = 800$ bits, $l_h = 8$ bytes; C: $l_v = l_d = 400$ bits, $l_h = 26$ bytes; D: $l_v = l_d = 400$ bits, $l_h = 8$ bytes.

Admittedly, the model which we used to draw these conclusions is too conservative for at least the following reasons:

1. The assumption that the maximum delay a voice packet encounters at a node is determined by the (non probabilistic) maximum possible queue length. This assumption affects n_v in region 2 of our performance curves.

2. The assumption that a voice packet experiences the maximum delay at each intermediate node. This assumption affects n_v in region 2 of our performance curves.

3. The assumption that the entire population of n_v active voice traffic sources are constantly generating traffic with rate V (no silent period). This affects regions 2 and 3 of our performance curves.

Regarding item 1 above, the deficiency of too conservatively combining the maximum queueing delay in a node is overcome in the model we present in the next section. However, we should point out that even with the improved model of next section, we find out that the number of voice channels, n_v, which the system can support is not large enough to refute our conclusions.

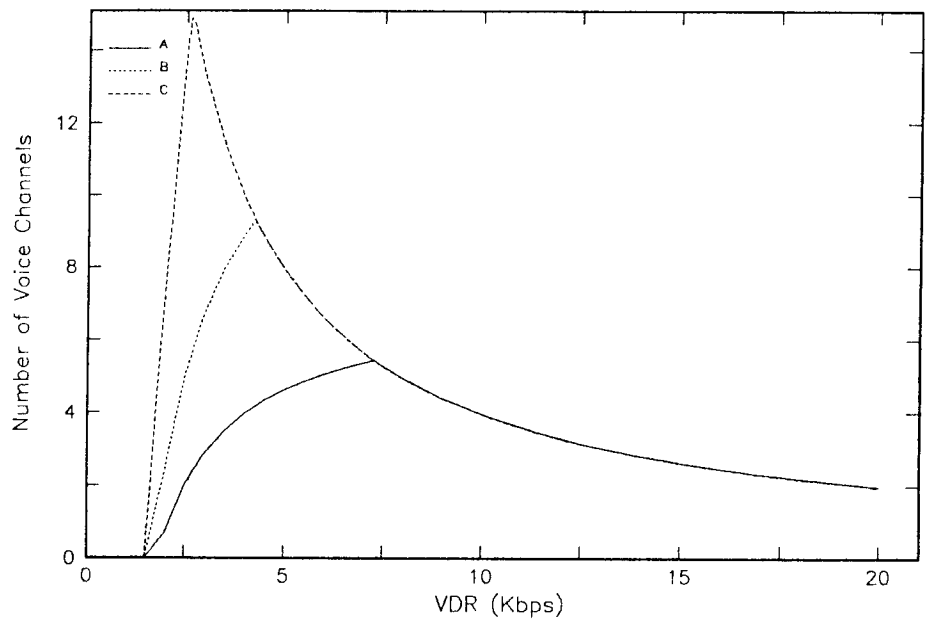

Figure 16. Number of voice channels vs. VDR for different β : $l_v = l_d = 800$ bits, $l_h = 26$ bytes, $D_{\max} = .5$ seconds, $n_h = 3$, A: $\beta = 1$, $C = 50$ kbps; B: $\beta = .5$, $C = 100$ kbps; B: $\beta = .2$, $C = 250$ kbps.

Regarding item 2, the chance that a voice packet experiences the maximum delay at each and every intermediate node depends on the path length, and in general is very small for long path lengths. In our numerical examples we have used $n_h = 3$ and for this path length the error introduced by our assumption is not too high.

Concerning the assumption of continuous talk spurts of all active users, it has been shown [2] that an active user in a conversation on the average, is silent 50 percent of the time. The possible multiplexing due to this property (assuming that the hardware is capable of detecting silent periods) affects our performance curves in region 2 and region 3. However, this multiplexing manifests itself only if the population s is very large. With the realistic channels capacities we have considered, the user population is too small for the multiplexing effect to take place and/or have significant effect.

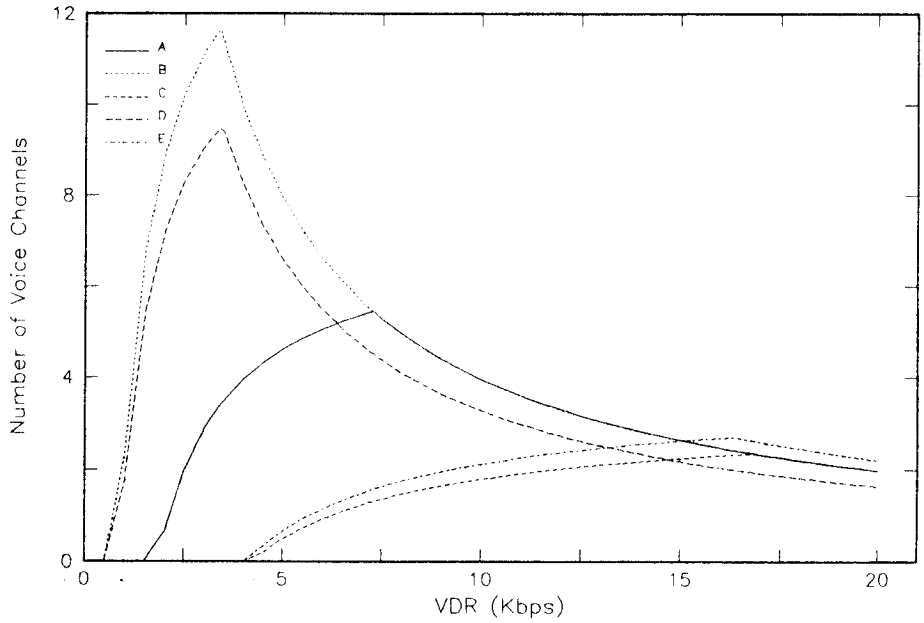

Figure 17. Number of voice channels vs. VDR for different maximum delays : C = 50 kbps, l_h = 26 bytes, β = 1, A: $l_v = l_d$ = 800 bits, D_{max} = .5 seconds; B: $l_v = l_d$ = 800 bits, D_{max} = 1 seconds; C: $l_v = l_d$ = 800 bits, D_{max} = .25 seconds; D: $l_v = l_d$ = 400 bits, D_{max} = .5 seconds; E: $l_v = l_d$ = 1600 bits, D_{max} = .5 seconds.

5. MORE ABOUT DELAY

As we pointed out at the closing of Section 4: "Delay Considerations", the main drawback of the model developed there (which we refer to as the "deterministic" model as opposed to the "probabilistic" model of this section) was the assumption that every voice packet encounters the maximum delay at all intermediate node. In this section we develop a more complex analytic model which overrides this assumption and provides us with a more accurate measure of the nodal delay. In the following, we first describe the development of the delay model for a single node. We then generalize this model for some nodes in tandem and present some numerical examples.

5.1 DESCRIPTION OF THE MODEL

Consider a communication node with M sources of voice traffic. (For the sake of brevity, we ignore the presence of data traffic. The effect of data traffic can be easily taken into account. Each source constantly generates voice signals which are first digitized at an A/D with rate V kbps and then assembled into packets of length l_v bits. With these assumptions, each source constantly delivers a packet to the node every $t_1 = l_v/V$ seconds. The (voice) packets are then multiplexed and sent over the outgoing channel of capacity C kbps; therefore, transmission time of a packet is $t_2 = l_v/C$ seconds. (Here we assume l_h, the header length, is zero; for non zero value of l_h we have $t_2 = (l_v + l_h)/C$.) The queueing discipline in front of the channel is FIFO. We refer to this queueing system as finite source periodic/D/1 system and we are interested in the distribution of the *maximum* queueing delay D.

To proceed with the analysis we consider a discrete time system in which each time unit, referred to as *slot*, is t_2 seconds. Therefore, each source periodically generates a packet every $N = t_1/t_2 = C/V$ slots. We refer to every N slots as a *frame* and it takes 1 slot to transmit a message out of the node. In the discrete time domain we assume all arrivals and departures occur at the beginning of a slot. Every source generates one and only one packet per frame. Within a frame, arrivals of packets from different sources is random; if more than one packet arrives in a slot, one of them is transmitted on the same slot (which one, is of no concern) and incurs a delay of 1, equal to its transmission time. The rest of packets are transmitted out on the succeeding slots.

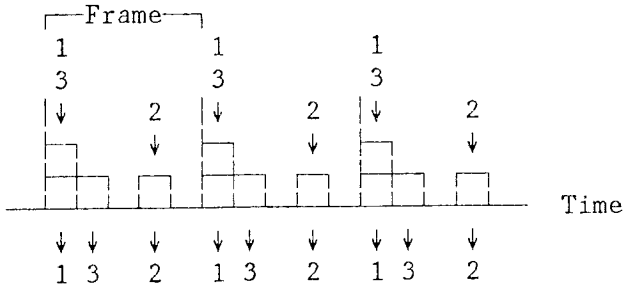

Figure 18. Arrivals and queue length for an environment with 3 sources and a frame size of 5.

Figure 18 shows an environment with M=3 traffic source and a frame size of N=5 slots. Sources 1 and 3 generate a packet at the first slot. One of them is sent out immediately (hence incurs a delay of 1 equal to its transmission time) and the other has to wait for one slot and then be transmitted out in the next slot (hence incurs a delay of 2). The packet from source 2 arrives in slot 4 and incurs no queueing delay and its total delay is just 1, its transmission time.

In order for the system to be stable, there cannot be more sources than the number of slots per frame, i.e., $M \leq N$. Another way to get this result is to note that the arrival rate of each source is $1/N$; there are M sources therefore the total arrival rate is M/N. For stability we require arrival rate not to exceed the service rate, hence $M/N \leq 1$. To study the maximum delay in a stable system under equilibrium, we need to observe the system only for one time frame; the starting slot of the frame does not affect our observation; this can be seen in Figure 18.

The analysis of the distribution of the maximum delay in such a system is presented in depth in Appendix C: "Distribution of Maximum Delay for Priodic/D/1 Queue". The analysis there provides us with a numerical procedure to derive the quantities of interest. Unfortunately, it turns out that the complexity of numerical computation is exponential with respect to M. Therefore, for values of $M>15$, we used a simulation model to find the distribution of maximum delay.

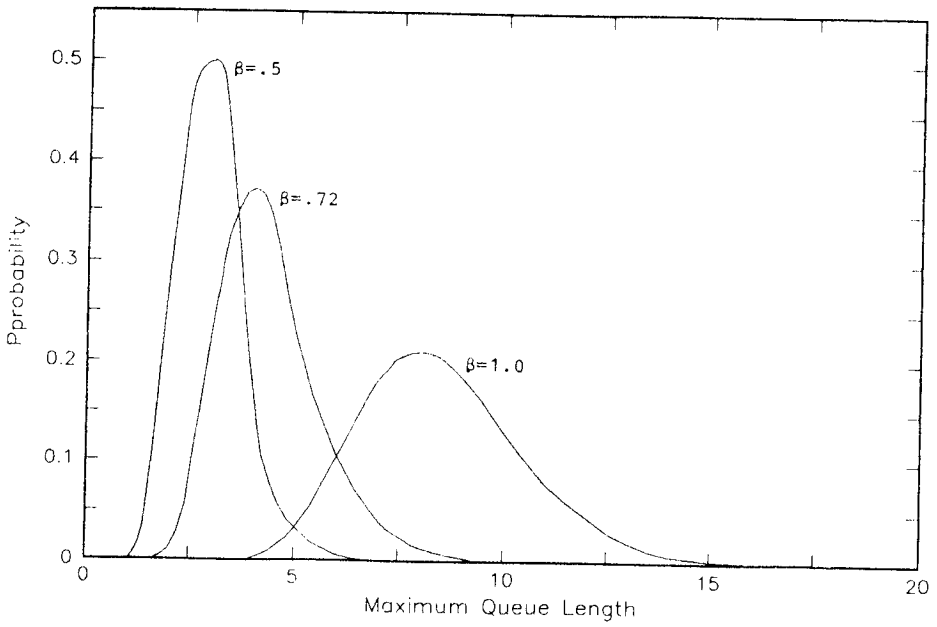

Figure 19. Distribution of maximum queue length : Frame length $N=50$ slots.

To give an example, consider a system with channel capacity of $C=100$ kbps, voice data rate of $V=2$ kbps, and packet length of 1000 bits. Therefore each frame has $N=C/V=50$. slots, each slot being 1000/100000 = .01 seconds. Figure 19 shows the distribution of maximum delay (in units of slots) for number of voice sources, M, equal to 25, 36, and 50. The curve for $M=25$ can be considered to represent an environment in which voice traffic can use maximum of 50% of the total capacity, i.e., $\beta=0.5$, similarly, $M=36$ represents the case of $\beta=.72$ and $M=50$ is for the case $\beta=1$.

Consider the curve for $M=50$. It is seen that the probability that $D=50$ slots (or $D=.2$ seconds) is quite negligible. In fact, the maximum delay is most likely to be 7 or 8 slots (or .07 or .08 seconds). $D=50$ is the value we used in our deterministic model of Section 4: "Delay Considerations" and it is clear how our new probabilistic approach provides us with a better performance measure.

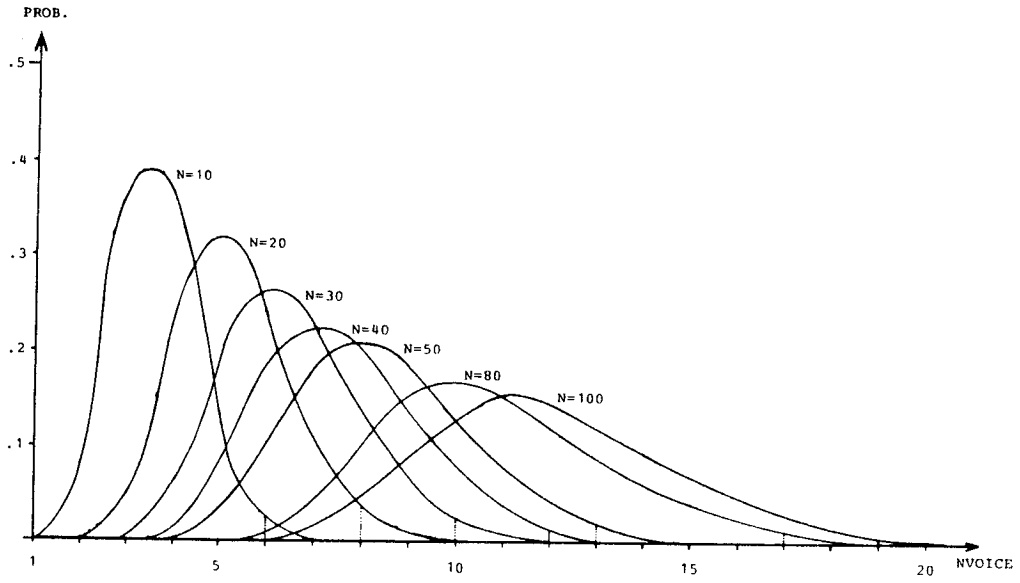

Figure 20. Density of maximum delay for different frame sizes : In all cases $M=N$.

Intuitively, when there is a large number of traffic sources (large M) the chance that all these sources generate packets at the same time is low. To demonstrate this effect more quantitatively we show in Figure 20 the distribution of delay in slots (not in seconds) for different values of N, the frame size. For all these curves we choose $M=N$. For example for the curve of $N=M=10$, the maximum delay can be 10 slots, and the peek of the distribution occurs at approximately 3. On the other hand for the curve of $N=M=100$, the peek occurs at a delay equal to 12 slots. Note that the chance that the maximum delay becomes greater than 25 is almost negligible.

Although Figure 20 demonstrates the advantage of a large population, it is not directly applicable for studying a real physical system. The reason is the following. Consider an environment where V = 10 kbps and l_v = 1000 (note that we assume l_h = 0). The curve for $M=N=100$ can be regarded as representing the case of $C=100$ kpbs (so that $N=C/V=100$). In this case the slot size is t_2 = 1000/1000000=.001 seconds. The same consideration for the curve $N=M=10$ shows that in this case $C=100$ kbps and the slot size is t_2 = 0.01. Therefore the slot size has different values for each case and the absolute value of the delay cannot be compared.

To get by this difficulty and still preserve the dimentionless property, we divide a time frame into 5 equal time intervals (note that the upper bound of the maximum delay is 1 frame), 0% to 20%, 21% to 40%, 41% to 60%, 61% to 80%, and 81% to 100% of a time frame. In Figure 21 we show for each value of $N=M$, the percentage of population with maximum delay in the specified range. For example, for $N=M=100$, all the packets suffer 1% to 20% of a time frame whereas for $M=N=40$, 72% of packets incur a maximum delay of 0%-20% of a frame and the remaining 28% incur a maximum delay of 21% to 40% of a frame time. This figure clearly demonstrates the advantage of a large number of voice traffic sources. Obviously, to be able to support a large population, the network should have large channel capacities. To demonstrate this point clearly, we present a numerical example.

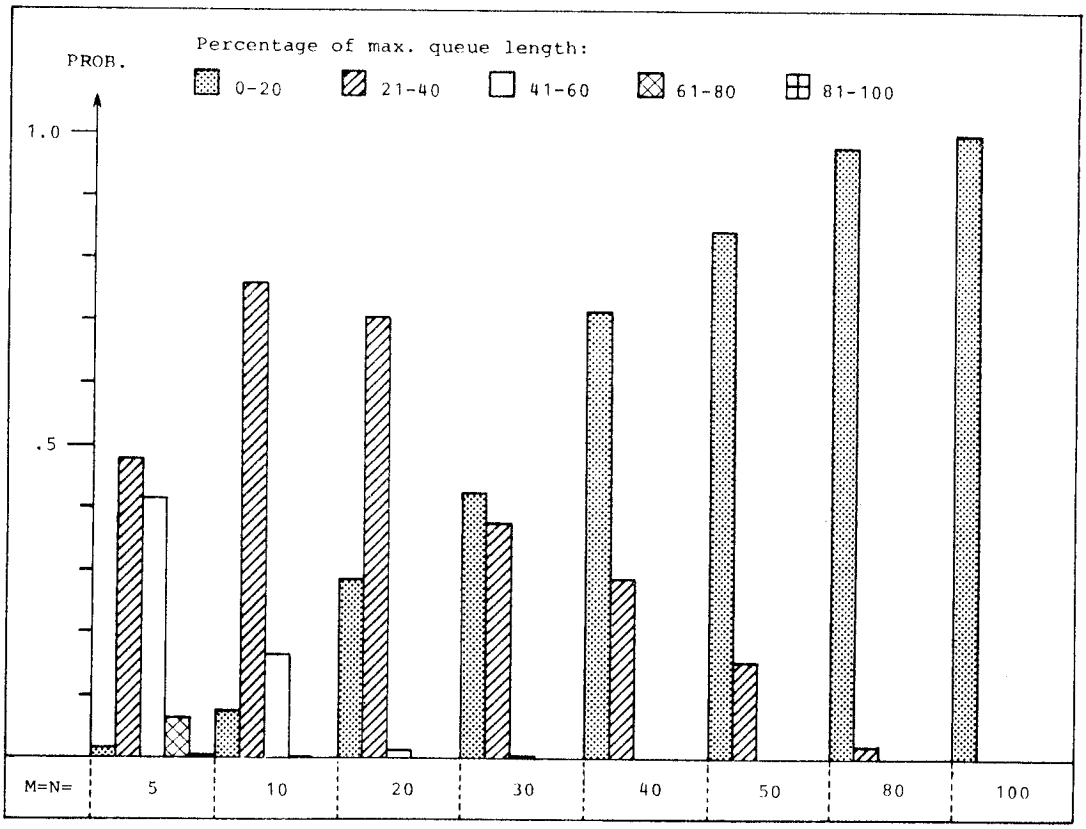

Figure 21. Delay incurred by percentage of traffic.

Assume $l_v = 1000$ bits and digitization rate $V=10$ kbps. In this case a frame length is $t_1 = .1$ seconds. Consider three channel capacities, $C=50$, 200, and 1,000 kbps, and in each case the number of voice sources is the maximum allowable, i.e., $N = 5$, 20, and 100, respectively (recall that $N=C/V$ if $l_h = 0$).

We use the corresponding column of Figure 20 to find out the percentile delay (measured in seconds) for each channel capacity; the result is shown in Figure 22.

Delay In Seconds	C=50kbps M=5 Percentage	C=200kbps M=20 Percentage	C=1000kbps M=100 Percentage
.000-.02	2	28	100
.021-.04	38	70	0
.041-.06	42	2	0
.061-.08	7	0	0
.081-.10	1	0	0

Figure 22. Percentile of maximum delay distribution.

In the deterministic model developed in Section 4: "Delay Considerations" we were concerned with the absolute maximum delay and we sized the network (i.e., specifying channel capacity, etc.) such that no voice packet suffers a delay more than a specified value which we called D_{max}. With the probabilistic delay of this section, rather than using the absolute delay, we can size the network such that only certain percentages of the traffic incur a given maximum delay. For example we may require that ε percent of traffic should have maximum delay of D_{max}. We refer to this value as the ε% delay. Such a goal can be achieved by using a density curve like the one shown in Figure 19

A question which may arise here is, how much do we gain (in terms of delay) by increasing the channel capacity. In Figure 23 we show the 99% delay (in terms of frame size) as a function of N=M. By virtue of the discussions for Figure 20 and Figure 22 there is a one to one correspondence between N and the channel capacity on one hand and the normalized delay (in terms of frame length) and the absolute delay (in seconds) on the other hand. Figure 23 shows that for too small C the delay is high. As we increase the capacity, there is a sharp decrease in delay, beyond which further increases in capacity does not decrease the delay by much. This point should be kept in mind in designing a network.

5.2 APPLICATIONS OF THE MODEL

In this section we use the probabilistic model to study the end-to-end delay and compare it with the results presented in Section 4: "Delay Considerations".

We start with underlying the assumptions. We consider a communication path of a network and assume the traffic at all nodes on the path to be identical. We assume this traffic is maximal, i.e., at each node all the active voice sources are generating talk spurt. However, in

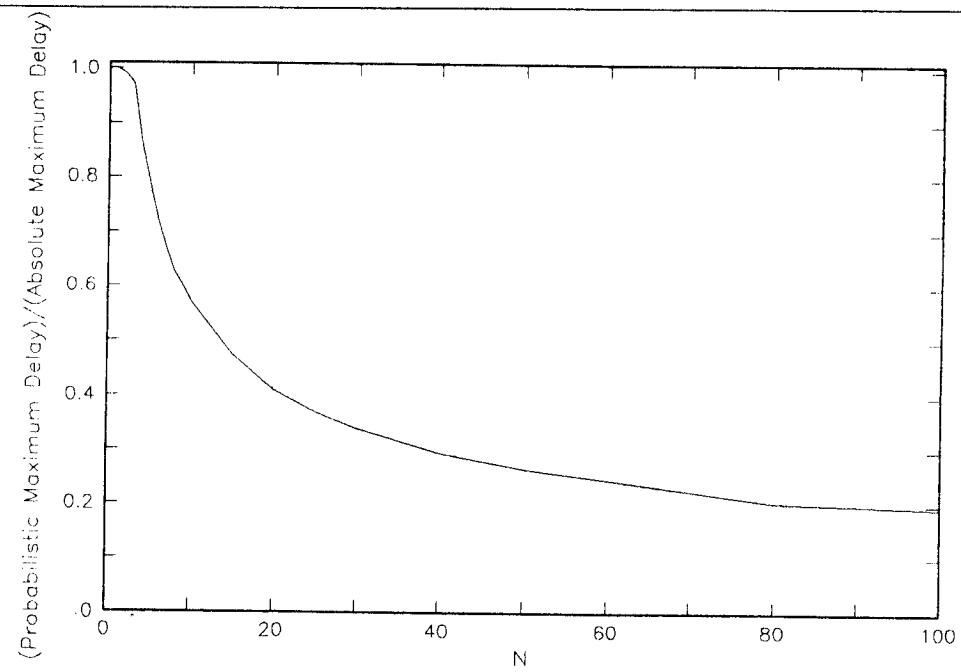

Figure 23. Relative Queueing Delay : ε=99%, β=1, N=M in all cases.

contrast to the deterministic model, we assume the arrival of messages from a traffic source to each node is random (in the sense of the discussion in Section 5.1: "Description of the Model"). Therefore, each node can be modeled as in Section 5.1: "Description of the Model" for which we can obtain the distribution of the maximum delay. As before, for a given environment (i.e. path length, voice packet length, etc.) and a given upper bound on the delay, D_{max}, we show the maximum number of voice channels, n_v, which the system can support as a function of voice data rate, VDR. As mentioned before, because we are dealing with a distribution, we also specify a percentage, ε, which specifies the percentage of traffic whose maximum delay should be less then D_{max}.

Figure 24 shows the number of voice channels as a function of VDR for different channel capacities C=50, 100, and 250 kbps. For each C we show the curves for the deterministic model (studies in Section 4: "Delay Considerations") and the probabilistic model. Note that in the area where the capacity is the bottleneck (region 3), the curves for both models coincide. Also, in region 1 (where VDR is insufficient) the curves overlap as well. The only difference is in region 2. As can be seen in the figure the deterministic model results in a more conservative value for n_v than the probabilistic model. The main point to observe in this figure is that for low channel capacities the difference between n_v as a result of the two models is marginal. Only when C is high does the difference become pronounced. We point out, however, that for high C the maximum value of n_v may achieved by using a VDR which is low. As we pointed out earlier, VDRs in this range are too expensive and may have poor quality (with present day technology).

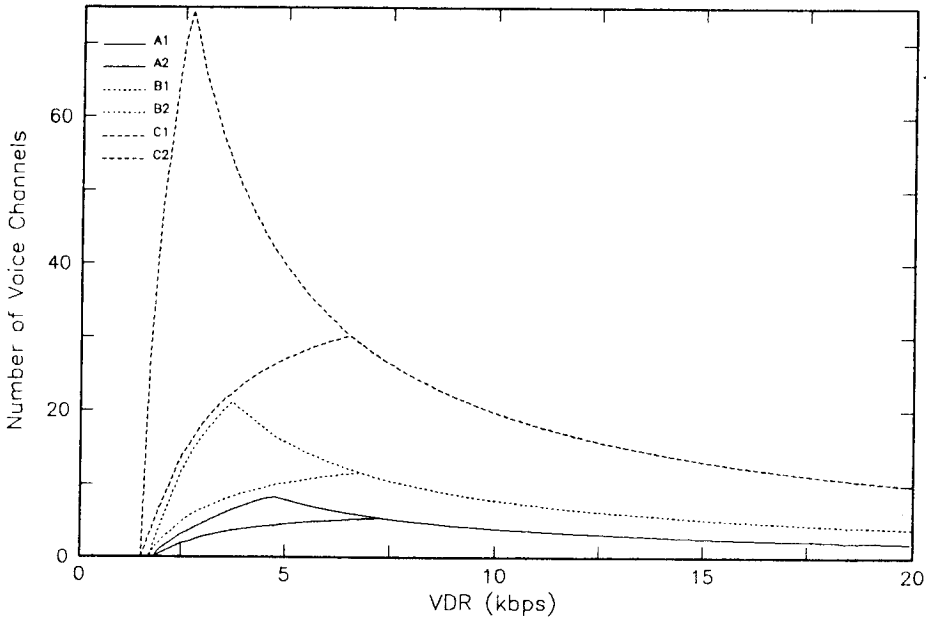

Figure 24. Number of voice channels vs VDR for different channel capacities : $l_v = l_d =$ 800 bits, l_h = 26 bytes, $D_{\max}$ = .5 seconds, n_h = 3, β=1, ε=99%, (A): C = 50 kbps; (B): C = 100 kbps; (C): C = 250 kbps.

6. PROCEDURES FOR GUARANTEED SERVICES

From the chapters before it is clear that the number of voice channels (sessions) has to be limited in order to guarantee a predetermined throughput with limited delay for each packet. In the following two procedures, referred to as "static" and "adaptive", which are suitable to establish voice sessions with guaranteed delay are proposed. More generally, the ideas are extendible to guarantee a certain level of service concerning a given criterion.

6.1 STATIC PROCEDURE

This procedure is simple and includes only little logical changes to existing store-and-forward networks except that a voice source has to request a connection to the destination and that such a request can be rejected even if connectivity is given.

6.1.1 Description of the Procedure

The basic idea of the algorithm is that a voice channel (session) is only accepted if for each node it is passing, a given maximum delay can be guaranteed. In order to be capable of making such a decision, each node has to maintain a Load Table (see Figure 25) which for each outgoing link contains the maximum number of voice calls (MAXVOICE) and the actual number of voice calls (ACTVOICE).

Link No.	MAXVOICE	ACTVOICE
1	M1	X1
2	M2	X2
3	M3	X3
.	.	.
.	.	.
l	Ml	Xl

Figure 25. Load Table for the static procedure.

The actual number of voice calls (or sessions) reflects the actual voice load of a link and changes dynamically according to that load. The maximum number of voice sessions is a constant for a given environment and is calculated according to our model algorithm in Section 5: "More About Delay" In 6.1.3: "An Example of Static Procedure" we give an example of this.

The procedure of establishing a voice session:

1. The source node sends a (voice) session request packet towards the destination.
2. Each intermediate node checks, by means of the Load Table, whether or not it can accept another voice session for the link it has to forward the packet.
   a. If it can accept the session, it forwards the request packet and reserves the capacity for it by incrementing ACTVOICE for the appropriate link.
   b. If it cannot accept an additional voice session, it does not forward the request packet, but sends a negative acknowledgment back to the source node.
3. Each node that the negative acknowledgment passes on its way back to the source cancels its reservation for the requested session (by decrementing ACTVOICE).
4. If the request packet reaches the destination node, a positive acknowledgment is sent back to the source node.

6.1.2 Discussion of the Procedure

A basic assumption required to make the described procedure applicable is that the network establishes a fixed path from source to destination for the duration of a session (unless line or node breakdown enforces rerouting).

In order to make the results from Section 5: "More About Delay" applicable for calculating MAXVOICE in the Load Table, a second assumption is that the environment harmonizes with the assumptions made there. This means that we assume a homogeneous system concerning voice data rate and voice packet length; there is no need to assume identical channel capacities for all links.

Advantages of the procedure:

1. It is simple to implement.
2. It creates almost no additional overhead.
3. It creates no additional network traffic if dynamic routing for session establishment is employed because each node is a self contained unit concerning the information it needs for its decision about acceptance of an additional voice session.

Disadvantages of the procedure:

1. It does not reflect the number of hops. Thus the calculations are based on an average number of hops (or an upper limit of hops for the majority of paths). If for an individual path the number of hops is different from this average, the guaranteed maximum delay is also different from the initially given delay. One possible refinement of the approach could come from introducing more than one priority class for voice according to the number of hops a path consists of. We did not work on this because the problem of having different path lengths is automatically solved with the adaptive algorithm introduced in Section 6.2: "Adaptive Procedure".
2. It assumes the single node worst conditions of Section 5: "More About Delay" for each node of a multinode path. This has the advantage that it automatically removes the problem of introducing some sort of information flow in the network concerning actual load conditions at different nodes. The disadvantage is that this assumption is fairly conservative in that the actual experienced delays may be significantly lower than the guaranteed ones. One reason is that in a multihop path, it is highly improbable that each node is working under worst-case conditions. The second reason is that in Section 5: "More About Delay" we assumed statistical arrival of packets. These statistical arrivals lead to queueing delays due to the necessary sequencing for transmission. This causes a subsequent node not to see the statistical arrival pattern for these packets, but to receive them sequentially, which - if this would be the only input stream - would give zero queueing delay. Thus, the situation is that in general a node sees one sequential input stream from each adjacent node and statistical arrivals only for those packets for which itself is the source node. In this case the delay most probably becomes less than what our procedure guarantees.

6.1.3 An Example of Static Procedure

We now present an example concerning how to achieve values for the MAXVOICE parameter in order to properly establish the Load Table. Assume the following environment:

Channel capacity of link l (C_l)	: 50 kbps
Voice data rate, VDR, (V)	: 4 kbps
Voice packet length (l_v)	: 800 bits
Header length (l_h)	: 200 bits
Maximum Delay (D_{max})	: .5 seconds

Furthermore, assume that the average path length is $n_h = 3$. From the voice data rate and the packet size we get *packetization delay*= $l_v/V = 0.2$ seconds.

Thus, there are .3 seconds left for transportation-oriented delays from the given .5 seconds maximum over all delay. As we assumed an average path length of 3 hops, we get *transportation delay per hop*=0.1 seconds.

A packetization delay of .2 seconds means that a voice digitization device will deliver a packet every .2 seconds and therefore the *frame length* (see Section 5: "More About Delay") is .2 seconds (only one voice packet of each voice channel can be contained in one frame). This frame is divided into 10 slots because the transmission of a single packet ($l_v + l_h$) across a 50 kbps link takes .02 seconds. The maximum transportation delay of .1 seconds is equal to 5 transmission times; thus, giving a maximum queue length of 5 packets. From this a nonprobabilistic analysis like that in Section 4: "Delay Considerations"would limit the maximum number of voice channels to be 5. However, the probabilistic approach of Section 5: "More About Delay" leads to a maximum population of 9.

Precisely it says for a population of 9 active voice sessions in an environment which - without delay restrictions - is capable of carrying 10 parallel voice sessions, the probability that *every* member of the population sees a delay of no more than 0.1 seconds (5 packet transmission times) is 0.993.

If we now view the environment slightly different and allow low priority data packets (of equal length) to interfere with voice packets (as we did in Section 4: "Delay Considerations"), the maximum queue length is reduced to 4 because voice traffic may be delayed by the transmission time of a data packet. In this case our investigations show that a population of 7 gives a maximum queue length of less than 4 transmission times with a probability of 0.994.

6.2 ADAPTIVE PROCEDURE

This procedure is a more sophisticated adaptive mechanism which, because of its adaptability to the dynamically changing network conditions, is expected to give better results compared

with the previous procedure. However, because of more a intelligent algorithm in network nodes, in terms of higher computational overhead in a node, it will be more costly to implement.

6.2.1 Description of the Procedure

The basic assumptions concerning the environment are very much the same as in Section 6.1: "Static Procedure". Again we assume a network with fixed paths for a session duration and again we assume homogeneity concerning VDR and packet length. We also state that voice traffic has priority over data traffic but we assume (and this is the basic new idea) one priority level for each voice channel instead of having only one priority level for the entire voice traffic. For this adaptive procedure, too, we have to specify a limit for the number of voice sessions per outgoing link (MAXVOICE) which in this case is identical to the number of different priority levels (for voice). In order to get a reasonable number for MAXVOICE, we again could apply the results from Section 5: "More About Delay"; however, in this case we need not have a delay oriented maximum and therefore also the maximum allowable number due to channel limitations would be a reasonable approach.

In this algorithm the maximum delay need not be a predefined network constant but can be individually specified for each request and is propagated with the session request packet. The information the nodes need to have to be capable of processing a request packet properly is contained in a priority-oriented delay table (PODT) for each outgoing link (see Figure 26).

Priority Level	Maximum Delay	Session Id.
1	DMAX1	
2	DMAX2	
3	DMAX3	
.	.	
.	.	
MAXVOICE	DMAXmaxvoice	

Figure 26. PODT: Priority Oriented Delay Table

The PODT has one entry for each (voice) priority level which contains the maximum delay (DMAX) a packet in this priority class may suffer. The problem of how to achieve these numbers does not touch procedural aspects and is therefore deferred to 6.2.3: "How To Achieve DMAXn in the PODT". The Session Identification entry of the PODT is initially empty and when operating contains a session identification of the session allocated to that priority level.

The session request packet has two fields: the MAXD field containing the submitted required maximum delay and the ACTD field which initially is empty and is used to accumulate the maximum delay guaranteed by the network.

The procedure of establishing a voice session:

1. The source node sends a request packet including the specification of the required maximum delay (MAXD field) towards the destination.
2. The source node and all intermediate nodes obtain the highest free priority level from their PODT table, assign it to the requested session and add the corresponding DMAX value to the ACTD field of the request packet.
3. If there is no priority level available (i.e. the new session would be in excess of the maximum number, MAXVOICE, of sessions) or if the ACTD field exceeds the MAXD field (i.e., delay requirements cannot be fulfilled) the node stops forwarding the request packet and sends a negative acknowledgement back to the source node.
4. The nodes which the negative acknowledgement has to pass on its way back to the source node cancel their reservation made for the requested session.
5. If the request packet reaches the destination, a positive acknowledgement is sent back to the source, which contains the MAXD field and the actual value of the ACTD field. (This can be achieved by simply returning the request packet.)
6. Each node the positive acknowledgement passes through, tries to reduce the previously assigned priority by some rules (there are several strategies thinkable including those establishing the notion of node to node fairness). The limiting condition is that the delay requirements are still met (i.e., ACTD may not exceed MAXD). If appropriate, the node corrects the session allocation in the PODT and the ACTD field accordingly and forwards the acknowledgement.

6.2.2 Discussion of The Procedure

Because the nodes have no knowledge about remaining path lengths (except the last intermediate node) or load conditions of other nodes, it is necessary to first assign the highest possible priority to a requested session in order to achieve a high probability that the sessions requirements can be met. This might result in much less delay than required thus guaranteeing better service than requested for. The price for this would be that the highest priority levels would always be occupied and new request would always be allocated the lowest priorities at each node, thereby continuously reducing the chance of matching the delay requirements. This would be still better than the static approach of Section 6.1: "Static Procedure" because the lowest priority here reflects the actual load condition of a node (instead of worst case condition). Nevertheless, this behavior is not desirable. The aim should be not to guarantee more than was requested for, thereby retaining as much as possible of the resources for other voice sessions. This is achieved by two measures. The first is that once a session is established the priority allocated cannot be changed. This is trivial concerning lower priorities because the guaranteed delay has to be met. But also higher priorities are not assigned even if such entries are becoming available by ending sessions. Note that as long as higher priority entries are free the service to lower priority sessions is the same as if priorities would have

been moved up; the important difference is that this level of service is not guaranteed because new sessions can be assigned to the free priority levels. The second measure is described in Step (6) namely, after it is clear that the delay requirements are met, beginning from the destination, intermediate nodes and the source node successively try to reduce the previously assigned (highest possible) priorities such that at the end of this process only the required delay is guaranteed.

Advantages:

The main advantage of this approach is its capability of adapting itself to different and dynamically changing conditions. These conditions contain dynamically changing load and different path lengths of individual sessions. Because it automatically makes use of advantageous conditions like short paths or low loaded nodes, we need not anxiously establish conservative limits concerning the maximum number of voice channels. In fact, by trying to give every session that and only that service, it needs this procedure normally allows low delay requiring sessions to be established even under bad conditions (high number of hops, high voice traffic at one or the other node) without necessarily reducing the population.

Disadvantages:

The implementation overhead and the computational overhead are higher than for the static approach. It should be possible to keep the computational delay during operation within limits because under normal operational conditions at most one packet for each priority level may be in a node at each point of time. Thus it is not necessary to maintain variable length queues for each priority level. Nevertheless the overhead is increasing with an increasing number of voice sessions which is normally a result of higher channel capacities. Thus for very high bandwidth the computational overhead of the nodes may become critical. On the other hand, the investigations of the previous chapters have shown, that delay problems become less severe with increasing bandwidth. So for very high bandwidth the main advantage of this scheme - allowing voice sessions under bad delay conditions without dramatically reducing the population - may be less important.

6.2.3 How To Achieve DMAXn in the PODT

In order for the advantages of the adaptive procedure to become efficient, it is necessary to provide reasonable values for the maximum delay field in the PODT. Our model from Chapter 5 cannot be used for this purpose. The basic assumption there was that the queue length a packet sees at its arrival time is a measure for the delay it will suffer. This assumption is valid for a one priority FIFO queue but not for a multi-priority (MP) environment.

As in the FIFO case, a deterministic and probabilistic estimate of the worst case delay will be given.

1. Deterministic worst case delay

   The basic idea is that a packet of priority level k can only be delayed by packets of higher priorities (1, ..., k-1). Thus in the worst case, when all packets of priority levels 1

through k-1 arrive together with level k, the packet of level k encounters a waiting time of k-1 transmission times giving a total (nonprobabilistic) maximum delay of k transmission times.

2. Probabilistic worst case delay

   In order to achieve numbers for the probabilistic worst case delays, the multi-priority environment was simulated. In the rest of this section we will elaborate on this approach.

The basic ideas of the model are exactly the same as described in Chapter 5. As in Chapter 5, we have the notion of a frame, periodicity within a frame length, the frame subdivided into slots (each representing one packet transmission time) and again we assume $M \leq N$ for system stability (M = number of voice sources). However, in this case not only the arrivals of packets are random (compare Figure 18) but in addition, we assign randomly (within a frame length) mutually exclusive priorities k ($k \varepsilon [1, ., M]$) to the arriving packets.

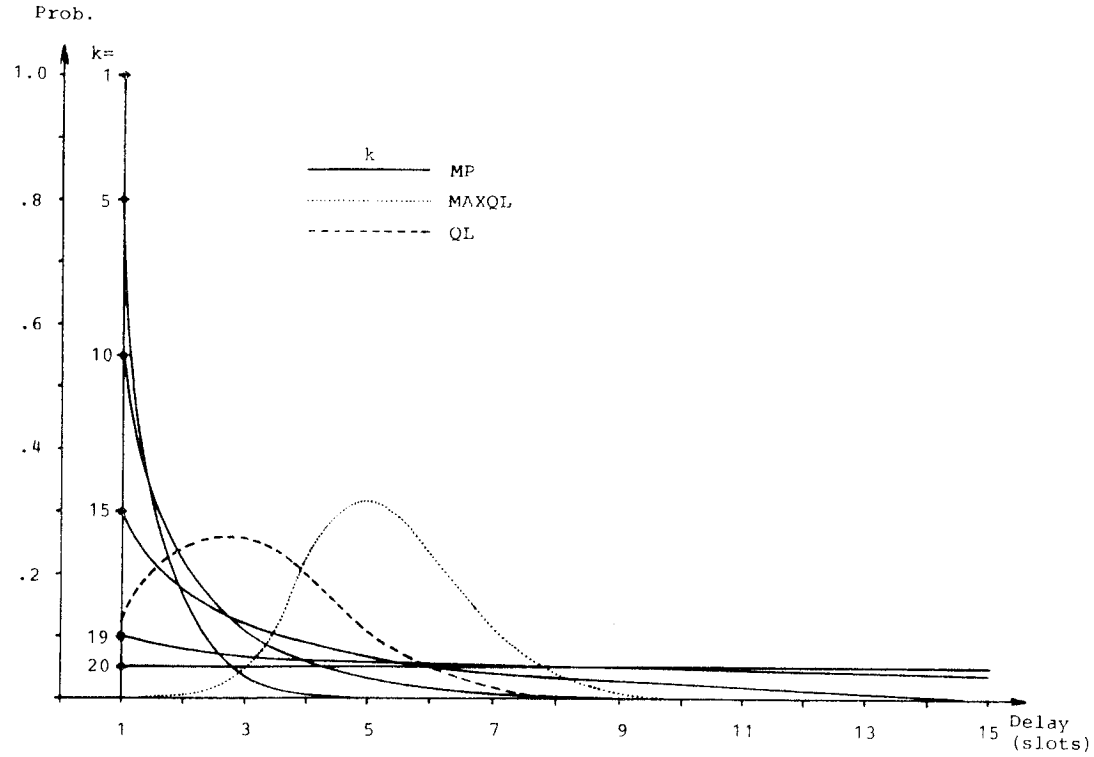

Figure 27. Density function of the maximum delay : $M=N=20$, and different k.

As a result, we get a probability distribution of delays for each priority level k. For $N=M=20$ and some priority levels k, these distributions are plotted in Figure 27. For comparison the maximum queue length (MAXQL) and the queue length (QL) distributions (FIFO) are added. The curves show that, except for k = 20 which is uniformly distributed, the distribution mass is biased to low delays, the most probable value being one which essentially means no

queueing delay. Indeed, there is a fairly high chance-especially of high priority packets-not to be delayed at all (e.g. priority level 1 packets will never be deferred, priority level 2 packets can maximally be deferred by one transmission time but only if they arrive together with priority level 1 packets, etc.).

On the other hand, the distributions show a fairly long tail due to the fact that especially low priority packets can be significantly deferred by simultaneously or successively arriving and bypassing higher priority level packets. This tail is one reason that the 99 percentile of the probabilistic approach gives only moderately better results than the deterministic approach (see Figure 28). The second reason for the relatively small difference between the deterministic and probabilistic approach is that the deterministic approach gives already fairly good results. Opposite to the nonprobabilistic FIFO scheme, which uniformly imposes the risk of suffering the maximum possible delay to any member of the population, the priority approach limits this risk to the lowest priority level, guaranteeing gradually better service (depending on the priority level) to anyone else. Having a small difference between the non-probabilistic worst case and the average is essential for real time environments where the former has to be the basis of the delay calculations.

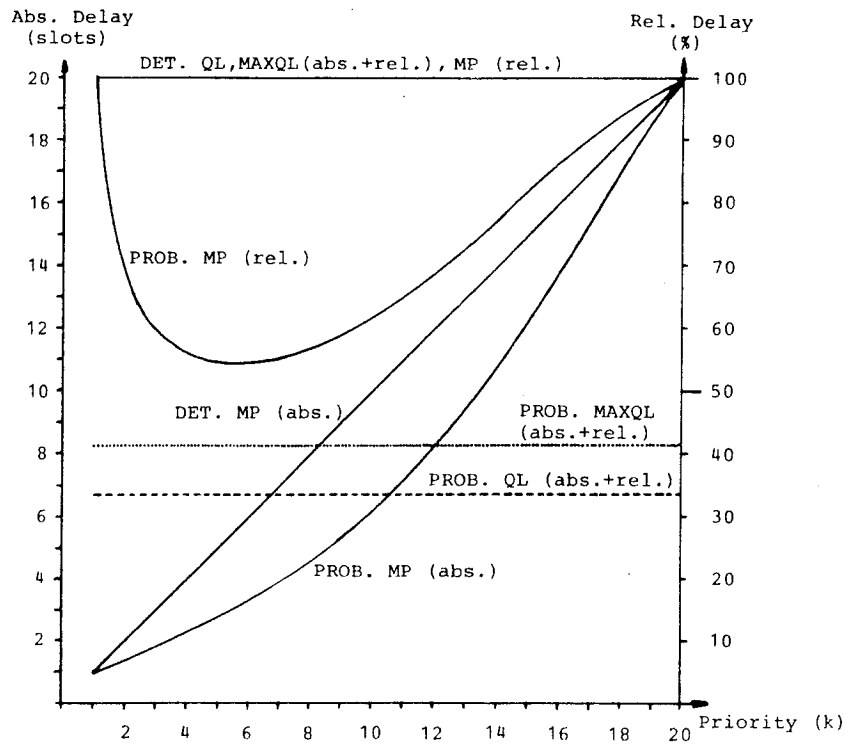

Figure 28. Relationship between deterministic and probabilistic worst case for FIFO and MP.

Achieving results for the cases of non-maximum population ($M<N$) is extremely simple in the multi-priority scheme. Because of the fact that the delay a packet of priority level k suffers is

by no means influenced by packets of lower priority levels, omitting these lower priority levels (and with that their packets) does not influence it either. This means that for any given N the delay behavior of a priority level k is independent of M as long as it exists ($k \leq M$).

The delay which in our model is given in transmission times, can easily be converted into real time and filled into the appropriate PODT entries if the environment (channel capacity, packet length, header length) is given.

7. CONCLUSIONS

In this report we studied the feasibility of, and some issues related to, transmission of packetized voice through existing store-and-forward data communication networks. We derived basic guidelines on the general design of a mixed voice/data communication network.

Regarding the issue of error checking we showed that it is infeasible to avoid error checking of voice packets in a system where both voice and data coexist. We also demonstrated that, in terms of transmission efficiency, there is no significant advantage in providing special, and more sophisticated, error handling procedures for voice packets. The same conclusion to some degree applies to the reduction of delay.

Our study of bandwidth and delay requirements indicate that channel capacity is the major obstacle in providing such a mixed service network. In contrast to the intuitive feeling that low data rate voice digitizers can be used to alleviate the need for high capacity channels, it turns out that low data rate digitizers, in conjunction with low channel capacities, cause the delay to be too high, hence this cannot be considered as a substitute for high bandwidth channels. Our analytic studies in Section 4: "Delay Considerations" and Section 5: "More About Delay" show that unless high bandwidth channels- and the nodal processing power to support them- are available, enhancing present data networks to provide packetized voice transmission is physically infeasible and economically unwise. The guidelines we provided in this report, nevertheless, can prove useful and valid when such physical obstacles are removed.

In the last section of this report we outlined two procedures to provide guaranteed services (more precisely guaranteed delay) in data networks. Guaranteed throughput/delay is not only required for packetized voice in a voice/data network, but is a feature which is desirable to be offered in most data networks. The materials of Section 6: "Procedures For Guaranteed Services" are an attempt in that regard.

APPENDIX A- PERFORMANCE OF TWO ERROR CHECKING SCHEMES.

In this appendix we present analytic models to determine transmission efficiencies of 1__CRC and 2__CRC error handling schemes. We will refer to these two schemes by method I and

method II, respectively (Figure 4). For method I retransmission occurs when CRC1 indicates an error. For method II CRC1 and CRC2 carry redundancy checks for the data field and the header, respectively. The operation of this scheme is explained in Section 3: "Error Handling". Briefly, if CRC2 indicates an error in the header field a retransmission occurs irrespective of the type of packet (voice of data) and/or the correctness of the data field (which is determined by CRC1). If CRC2 is correct but CRC1 indicates an error in the data field then if the packet is a voice packet, no retransmission is necessary. However, in the case of a data packet, a retransmission is requested.[3] It should be noted that retransmitted packets are always data packets.

Let

l_h = length of header (bits)
l_m = length of data field (bits)
c_1 = length of CRC1 (bits)
c_2 = length of CRC2 (bits)

Furthermore, let

$$\begin{aligned} L_1 &= l_h + l_m + c_1 && \text{packet length; method I} \\ L_2 &= l_h + c_2 && \text{header length plus CRC; method II} \\ L_3 &= l_h + l_m + c_1 + c_2 && \text{packet length; method II} \\ L_4 &= l_m + c_1 && \text{length of data field plus CRC; method II} \end{aligned} \tag{A.1}$$

The error rate of the communication link is represented by P_e, i.e.

$$P_e = \Pr[\text{a bit in error}]$$

and we define

$$P_i \triangleq \Pr[\text{a string of length } L_i \text{ bits contains at least one bit in error}] \quad i = 1, 2, 3, 4$$

P_i is given by

$$P_i = 1-(1-P_e)^{L_i} \quad i = 1, 2, 3, 4 \tag{A.2}$$

We use α to denote the fraction of traffic which is data packet, i.e.

$$\alpha = \Pr[\text{a packet is a data packet}]$$

and of course (1-α) is the probability that a packet is a voice packet. Finally we use $\bar{n}_{\text{I}}$ and $\bar{n}_{\text{II}}$ to denote the average number of retransmissions (including the original transmission) of methods I and II, respectively and e_{I} and e_{II} to denote the transmission

---

[3] It is assumed that the header carries information regarding the type of packet, voice or data.

efficiencies of these two methods.

Method I. Using a renewal type argument [5], the average number of transmissions for method I is given by

$$\bar{n}_1 = (1 + \bar{n}_1)P_1 + (1-P_1)$$

Therefore,

$$\bar{n}_1 = \frac{1}{1-P_1}$$

and we have

$$e_1 = \frac{l_m}{\bar{n}_1 L_1}$$

After substitution and some algebra, we get

$$e_1 = \frac{(1-P_e)^{l_h+l_m+c_1}}{l_h+l_m+c_1} l_m \tag{A.3}$$

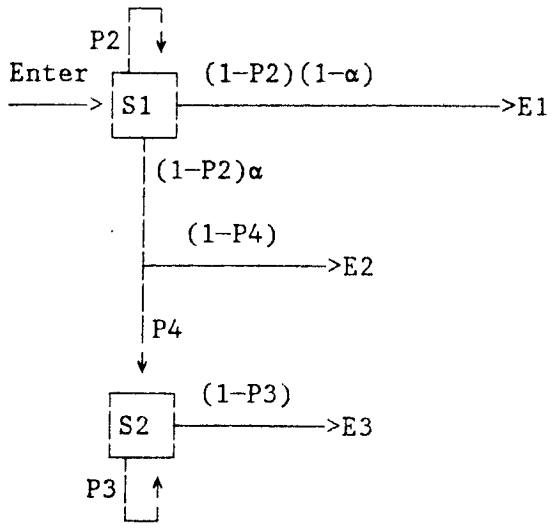

Figure A.1. State diagram for the 2__CRC scheme.

Method II. The automaton representing state transition and operation of the system under this method is shown in Figure A.1. The system starts in state S1. If the header has an error (with probability P_2), then the packet has to be retransmitted. If there is no error in the header and the packet is a voice packet, then the transmission is complete, exit E1 in Figure A.1. If the header is correct and the packet is a data packet (with probability $(1-P_2)\alpha$) and there is no error in the data field (with probability $1-P_4$), again the transmission is complete, exit E2 in Figure A.1. Otherwise, the system reaches state S2 and the data packet has to be retransmitted. Once the system reaches state S2, in order the transmission to be complete the entire packet of length $L_3 = l_h + l_m + c_1 + c_2$ should be received error free; this occurs with probability $(1-P_3)$.

Let $\bar{n}'_{II}$ be the average number of times the system visits state S2, then we have

$$\bar{n}_{II} = (1 + \bar{n}_{II})P_2 + (1-P_2)\{(1-\alpha) + \alpha[(1-P_4) + (1 + \bar{n}'_{II})P_4]\} \tag{A.4}$$

and we have

$$\bar{n}'_{II} = (1 + \bar{n}'_{II})P_3 + (1-P_3)$$

therefore,

$$\bar{n}'_{II} = \frac{1}{1-P_3} \tag{A.5}$$

Using the values of $\bar{n}'_{II}$ in Eq. (A.5) and after some algebra, we get

$$\bar{n}_{II} = \frac{(1-P_3) + \alpha P_4(1-P_2)}{(1-P_3)(1-P_{2)}} \tag{A.6}$$

The transmission efficiency e_{II} is given by

$$e_{II} = \frac{l_m}{\bar{n}_{II}L_3}$$

and after proper substitution we finally get

$$e_{II} = \frac{(1-P_e)^{l_m+c_1} + \alpha[1-(1-P_e)^{l_m+c_2}]}{(1-P_e)^{l_h+l_m+c_1+c_2}} \tag{A.7}$$

APPENDIX B- MAXIMUM DELAY UNDER DETERMINISTIC ASSUMPTIONS

In this appendix an analytic model to calculate the maximum delay a voice packet may encounter in passing through a communication path of a packet switched network is developed. The following notations will be used in the model:

V VDR (voice data rate)
C Channel capacity (between nodes)
β Fraction of Channel capacity used by voice
D Maximum delay
l_v Voice packet length
l_d Data packet length
l_h Packet Header length
n_h Number of hops on the communication path
n_v Number of voice channels

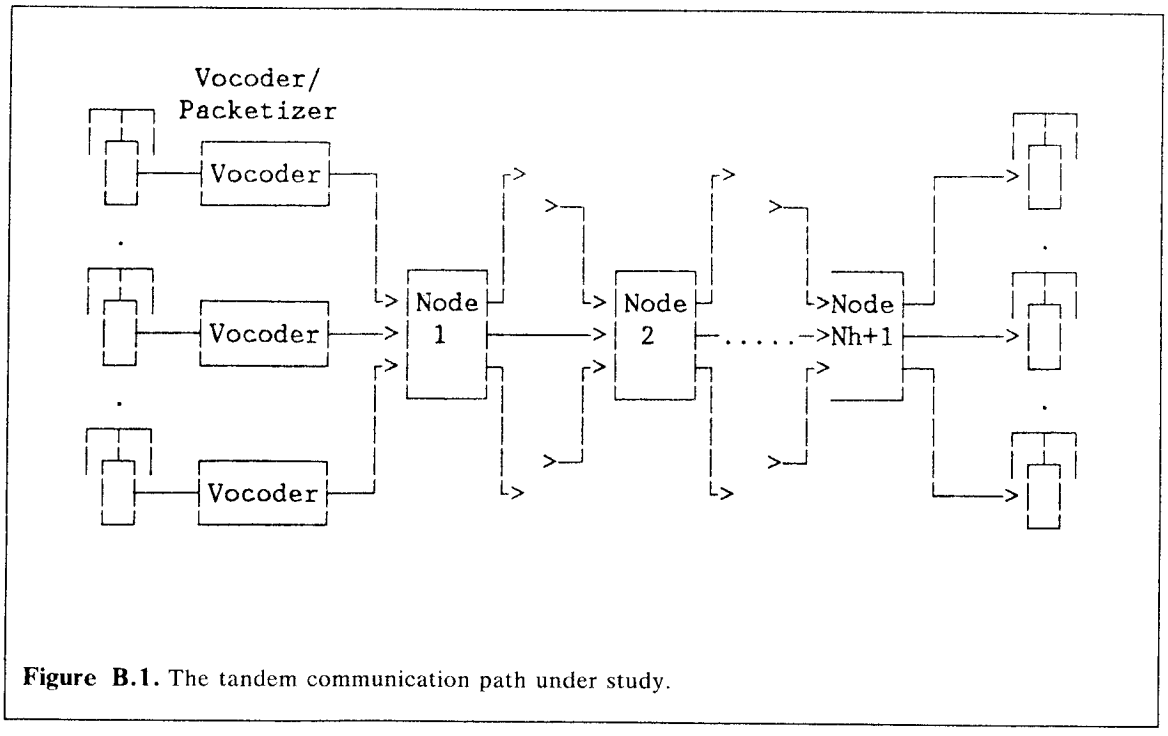

Figure B.1. The tandem communication path under study.

We consider a communication path of n_h hops with single links of capacity C between every pair of adjacent nodes, Figure B.1. In addition to data traffic, which consists of packets of length l_d, there are n_v voice channels active at every node. Voice traffic is generated by digitizers of rate V; the voice traffic is packetized into packets of length l_v. We assume the load at all nodes are equal. Furthermore, the assumption is that voice traffic has nonpreemtive priority over data traffic.

When there are n_v voice channels active, the maximum delay is encountered when all voice channels are generating data at their peek. Therefore each source submits a packet every l_v/V seconds. In this situation when a voice packet enters a node and joins a queue for transmission it faces $(n_v - 1)$ voice packets belonging to the other voice channels in front of it. In addition, there may be a data packet which has just started transmission out of the node.

With the above considerations we can find the maximum delay which consists of the packetization delay, the transmission delay, and the queueing delay. The packetization delay is simply l_v/V. The transmission delay at a node is given by $(l_v + l_h)/C$, because a header of l_h is appended to every packet. Therefore, the total transmission delay is given by

$$\frac{l_v + l_h}{C} n_h$$

The queueing delay at each node is given by

$$\frac{l_d + l_h}{C} + (n_v - 1)\frac{l_v + l_h}{C}$$

The first component is the maximum delay due to the data packet which is in the process of transmission. The second component is the delay due to packets of the other $(n_v - 1)$ voice channels. The total queueing delay is n_h times this delay.

Summing up, we have the following expression for the maximum delay when there are n_v voice channels active.

$$D = \frac{l_v}{V} + \frac{l_v + l_h}{C} n_h + [\frac{l_d + l_h}{C} + (n_v - 1)\frac{l_v + l_h}{C}]n_h$$

which after simplification reduces to

$$D = \frac{l_v}{V} + [\frac{l_d + l_h}{C} + n_v\frac{l_v + l_h}{C}]n_h \qquad \text{(B.1)}$$

Because only β fraction of the capacity C can be used by voice, we have the following condition which should be satisfied as well

$$n_v V \leq \beta C \qquad \text{(B.2)}$$

Note that when $\beta=1$, then Eq. (B.2) gives the stability condition (arrival rate $\leq$ service rate). We can used Eq. (B.1) and Eq. (B.2) to derive the maximum possible delay, i.e., the delay when the number of voice channels is

$$n_v = \frac{\beta C}{V} \qquad \text{(B.3)}$$

This delay is given by

$$D = \frac{l_v}{V} + [\frac{l_d + l_h}{C} + (\frac{\beta C}{V})\frac{l_v + l_h}{C}]n_h \qquad (B.4)$$

APPENDIX C- DISTRIBUTION OF MAXIMUM DELAY FOR PERIODIC/D/1 QUEUE

In this appendix we analyze the distribution of maximum delay for the Periodic/D/1 queue introduced in Section 5: "More About Delay".

Consider a discrete time single server queue with M sources of traffic in which each source sends one and only one packet every N time unit and it takes 1 time unit to transmit a packet out of the node. In our discussion we refer to a time unit as a *slot* and to every N time units as a *frame*. We assume all departures and arrivals occur immediately *before* and *after* the beginning of a time slot, respectively. If more than one arrival occurs at the beginning of a slot, one of them (which one, is of no matter, but see below) is sent out in that slot and the remaining packets are sent out in a succeeding slot (we will refer to these remaining packets as *carry over* packets). That is to say, the intra slot queueing discipline is random. However, packets which are carried over to a slot have priority over the newly arrived packets; i.e. the inter slot discipline is FIFO.

To formalize the idea let us assume that slots are numbered consecutively 1, 2, ... Furthermore, assume that at slot 1 the system is empty. Define

$n_i \stackrel{\Delta}{=}$ number of packets which arrive at the beginning of slot i
$S_i \stackrel{\Delta}{=}$ Maximum delay of packets *arriving* in slot i

Then we have

$$\begin{aligned} S_1 &= n_1 \\ S_2 &= n_2 + (S_1 - 1)^+ = n_2 + (n_1 - 1)^+ \\ S_3 &= n_3 + (S_2 - 1)^+ = n_3 + [n_2 + (n_1 - 1)^+ - 1]^+ \end{aligned}$$

and in general

$$S_j = n_j + (S_{j-1} - 1)^+ \qquad (C.1)$$

where $(i)^+ \stackrel{\Delta}{=} max\{i,0\}$. We are interested in maximum of S_i, that is

$$S^{\max} \stackrel{\Delta}{=} \max_{1 \le i \le N} \{S_i\} \qquad (C.2)$$

The example in Figure 18 helps clear the idea.

As we mentioned in Section 5: "More About Delay", to study the maximum delay we have to study only the arrivals within in a frame. The starting point of a frame is not important. For example in Figure 18, no matter where we start the frame, the maximum delay is 2. The important point to notice is that for any frame of length N the number of packets carried in the frame is equal to the number of packets carried out out of the frame.

Assuming that the arrival of a packet in a frame is random (i.e. a packet may occupy any slot of a frame), then the problem at hand is very similar to the combinatorial problem of placing M balls in N urns. The main difference between the two problems is that in the latter case there is no dependency between the urns (slots); however, because of the *carry over* phenomenon, the waiting time for a slot depends on the history of arrivals (assignment) in the previous slots.

Assume we randomly assign packets to the slots. In general when there are m packets and n slots the number of different assignments is n^m. Let

$F_n^m(k) \triangleq$ Probability of having a maximum delay of k or less in assigning m packets into n slots,
$k \leq m \leq n$

Then we have

$$\Pr[S^{\max} = K] = F_N^M(K) - F_N^M(K-1) \qquad K \leq M \leq N \tag{C.3}$$

We now develop an algorithm to find the probabilities $F_n^m(k)$. To do this, we first impose a condition on number of packets which are carried into (and out of) a frames. Let

$F_n^m(k\&l) \triangleq$ Prob. that m packets are assigned into a frame of n slots, there are l packets carried into the frame and the delay is k or less,
$l < k \leq m \leq n$

Then we have

$$F_n^m(k) = \sum_{l=0}^{k-1} F_n^m(k\&l) \tag{C.4}$$

To find quantities $F_n^m(k\&l)$ we define

$L_n^m(k,l_1,l_2) \triangleq$ Number of times m packets are assigned into a frame of n slots, carry in is l_1, carry out is l_2 and maximum delay is k or less,
$l_1, l_2 < k \leq m \leq n$

then we have

$$F_n^m(k\&l) = \frac{L_n^m(k,l,l)}{n^m} \tag{C.5}$$

and

$$F_n^m(k) = \frac{\sum_{l=0}^{k} L_n^m(k,l,l)}{n^m} \tag{C.6}$$

The quantities $L_n^m(k,l_1,l_2)$ can be calculated by allocating 1, 2, .. $k-l_1$ packets to the first slot and finding the number of ways the rest of packets can be assigned to the remaining n-1 slots and the conditions are satisfied. That is

$$\begin{aligned} L_n^m(k,l_1,l_2) = {} & L_{n-1}^m(k,l_1-1,l_2) \\ & + C(m,1)L_{n-1}^{m-1}(k,l_1,l_2) \\ & \cdot \\ & + C(m,i)L_{n-1}^{m-i}(k,l_1+i-1,l_2) \\ & \cdot \\ & + C(m,k-l_1)L_{n-1}^{m-k+l_1}(k,k-1,l_2) \end{aligned}$$

or

$$L_n^m(k,l_1,l_2) = \sum_{i=0}^{k-l_1} C(m,i)L_{n-1}^{m-i}(k,l_1+i-1,l_2) \tag{C.7}$$

We can now use Eq. (C.7) in Eq. (C.6) and Eq. (C.3) to find the desired distribution.

REFERENCES

[1] Bially, T., B. Gold and S. Seneff, "A Technique for Adaptove Voice Flow Control in Integrated Packet Networks," *IEEE Trans. on Comm.*, Vol. COM-28, No. 3, Match 1980, pp. 325-333.

[2] Brady, P.T., "A Statistical Analysis of On-Off Patters in 16 Conversations," *BSTJ*, Vol. 47, No. 1, Jan. 1969, pp. 73-90.

[3] Cotton, I.W., "Making Machines Talk: Simulated Speech, Part One," *Data Communications*, Jan. 1981, pp. 75-80.

[4] Coveillo, G.J., O.L. Lake and G.R. Redinbo, "System Design Implications of Packetized Voice," *Proc. of ICC 77*,Chicago, Ill. 1977, pp. 38.3-49 to 38.3-53.

[5] Cox, R.D., *Renwal Theory*, Butler & Tanner Ltd, Frome and London, 1962.

[6] Forgie, J.W., "Speech Transmission In Packet-Switched Store-and-Forward Networks," *Proceedings of National Computer Conference*, June 1975, pp.137-142.

[7] Gitman, I. and H. Frank, "Economic Analysis of Integrated Voice and Data Networks: A Case Study," *Proceedings of The IEEE*, Vol. 66, No. 11, November 1978, pp. 1549-1570.

[8] Gold, B., "Digital Speech Networks," *Proceedings of the IEEE*, Vol. 65, No. 12, Dec. 1977, pp. 1636-1658.

[9] Kermani, P. and L. Kleinrock, "A Tradeoff Study of Switching Systems in Computer Communication Networks," *IEEE Trnas. on Computers*, Vol. C-29, No. 12, December 1980, pp. 1052-1060.

[10] Occhiogrosso, B., "Digitized Voice Comes Out Of Age, Part 2- Techniaues," *Data Communications*, April 1978, pp. 63-81.