October 1981 LIDS-P-1153

# FLOW CONTROL AND ROUTING TECHNIQUES FOR INTEGRATED VOICE AND DATA NETWORKS†

by

Oliver C. Ibe*

## ABSTRACT

We consider a model of integrated voice and data networks. In this model the network flow problem is formulated as a convex optimization problem. The objective function comprises two types of cost functions: the congestion cost functions, which limit the average input traffic to values compatible with the network conditions; and the rate limitation cost functions, which ensure that all conversations are fairly treated. A joint flow control and routing algorithm is constructed which determines the routes for each conversation, and effects flow control by setting voice packet lengths and data input rates in a manner that achieves optimal tradeoff between each user's satisfaction and the cost of network congestion. An additional congestion control protocol is specified which could be used in conjunction with the algorithm to make the latter respond more dynamically to network congestion.

---

†This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by ARPA under Grant No. N00014-75-C-1183.

*Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.

## I. Introduction

The factors which favor the integration of voice and data onto one network include the expected cost savings to be derived from sharing switching and transmission facilities, and the promise of using network resources more efficiently [1] - [3]. Three switching techniques have been proposed for integration. These are circuit switching, hybrid switching, and packet switching. In circuit switching, a fixed capacity end-to-end circuit is established for a pair of voice or data users before they commense their conversation. Thus no queueing delays are encountered at the nodes, and the end-to-end transmission facilities are dedicated to the users for the duration of their conversation. In packet switching, both voice and data conversations are digitized and segmented into packets and routed in a store-and-forward manner [4], [5]. In hybrid switching, each channel is partitioned into two subchannels: a circuit-switched subchannel, which accommodates voice traffic; and a packet-switched subchannel, which accommodates data traffic. To increase the channel utilization, a "movable boundary" feature can be incorporated which permits data packets to use any residual circuit-switched capacity that may be momentarily available due to voice traffic variations [3].

By handling voice and data traffics in an essentially uniform fashion, packet switching provides the capability for voice and data conversations to respond automatically and rapidly to changes in traffix mix. Furthermore, the success of such public packet-switched data networks as the TELENET [6], the TYMNET [7], the TRANSPAC [8], and the DATAPAC [9] has given greater impetus to the implementation of packet-switched integrated networks. However, voice conversations are very sensitive to delay variations while packet switching systems exhibit variable delay due to the possible queueing of packets at the different nodes. Therefore, before voice and data can be successfully integrated onto one packet-switched network a number of issues have to be resolved. In this paper we consider two such issues; namely, flow control and routing.

By flow control we mean the set of mechanisms used to regulate the entry of traffic into the network. The main functions of flow control in a packet network include the prevention of throughput degradation and loss of efficiency due to overload; deadlock avoidance; fair allocation of network resources; and matching the rate at which the network accepts packets to the rate at which these packets are generated [10]. Voice and data traffics make different and conflicting demands on the network: they show different tolerances to delay and errors. Voice conversations require continuous and real-time delivery; they are very sensitive to delay but less sensitive to errors. Data, on the other hand, are generally intolerant of errors but less sensitive to delay; they need to be reliably delivered to their destinations. Thus flow control should perform two additional functions in a packet-switched integrated network: It should ensure that voice packets are transmitted with essentially constant delay while trading the speech quality in response to network conditions, and that data packets are transmitted with maximum reliability (i.e. no error or lost information) with delay as a secondary issue.

A reasonable flow control protocol for an integrated packet network would then be one that maintains speech continuity under all network conditions while degrading the voice quality, by reducing the voice bit rate, when the network becomes congested. This is precisely what the embedded coding scheme

[11] does. However, by our definition the embedded coding scheme does not by itself constitute a flow control scheme. The flow control scheme that has been proposed for use in conjunction with the embedded coding scheme has been shown to have the potential for unstable operation [12].

In this paper we present a model of the integrated voice and data packet network (Section II). The work reported in [11] was instrumental to our formulating the voice traffic model presented in this paper. A joint flow control and routing algorithm is constructed which uses short term average information on the network utilization to effect flow control and to determine the routes for each conversation (Section III). Flow control is effected by setting the voice packet lengths and the data input rates in a manner that achieves optimal tradeoff between each user's satisfaction and the cost of network congestion. However, even though our flow control scheme is not prone to unstable operation if the scale factors are well chosen, the joint flow control and routing algorithm is not as dynamic as the embedded coding scheme in responding to network congestion. To make the algorithm respond more dynamically to network congestion, we develop a modified embedded coding scheme that could be used in conjunction with the algorithm (Section IV). An important feature of our scheme is that, unlike [11], the interaction between voice and data traffics is taken into consideration in constructing the algorithm. Also our model has the capability to handle a traffic structure that comprises different traffic classes which have different levels of delay sensitivity. Details of this feature are given in [14].

## II. Network Model

We consider a network consisting of N nodes, denoted by 1, 2,..., N; and L directed links, where a link that goes from node k to node $\ell$ is denoted by $(k,\ell)$. The set of links is denoted by $L$. We assume that if link $(k,\ell)$ exists then link $(\ell,k)$ also exists. We also assume that both voice and data sources can be modelled as random processes with slowly varying input traffic statistics. Further details of the motivation for modelling data sources this way can be found in Gallager [15]; we concentrate on the discussion of voice sources.

When a speaker is off-hook (i.e. in the conversational mode), he alternates randomly between the talkspurt and silence modes. When in talkspurt, he generates speech which is packetized by a voice digitizer and sent into the network at a fixed rate of $\beta$ packets/sec. The quality of a voice conversation depends on the bit rate. Since we have fixed the voice packet rate, speech quality then depends on the length of the voice packets; the longer the packets, the higher is the speech quality. We let the speech quality (and hence the voice packet lengths) vary in accordance with the network conditions. For data, we let the permisible input rate vary in accordance with network conditions and, therefore, we assume that data packet lengths are independent of network control. We use the following notations:

(i,j) conversation = a conversation that enters the network at node i and is destined for node j.

$O(i)$ = the set of nodes k for which $(i,k)\in L$

$I(i)$ = the set of nodes k for which $(k,i)\in L$

$f_{ik}^{v}(j)$ = the expected voice traffic on link (i,k), in bits/sec, destined for node j, where averaging is over the duration of a conversation

$f_{ik}^{d}(j)$ = the expected data traffic on link (i,k), in bits/sec, destined for node j, where averaging is over the duration of a data session

$\ell_{ij}$ = the expected packet length, in bits, of (i,j) voice conversations

$r_{ij}$ = the expected input rate, in bits/sec, of an (i,j) data session

$\beta$ = the rate at which voice packets are emitted by a voice digitizer during a talkspurt, in packets/sec

$F_{ik}^{v} = \sum_{j=1}^{N} f_{ik}^{v}(j)$, the expected aggregate voice traffic on link (i,k)

$F_{ik}^{d} = \sum_{j=1}^{N} f_{ik}^{d}(j)$, the expected aggregate data traffic on link (i,k)

$F_{ik} = F_{ik}^{v} + F_{ik}^{d}$, the expected total traffic on link (i,k)

$C_{ik}$ = the capacity of link (i,k), in bits/sec.

$m_{ij}$ = the number of (i,j) data sessions

$n_{ij}$ = the number of (i,j) "off-hook" speakers

$n_{ij}^{T}$ = the number of the $n_{ij}$ off-hook speakers in talkspurt

$n_{ij}^{S}$ = the number of the $n_{ij}$ off-hook speakers in silence

Usually $n_{ij}^{T}$ changes so fast that it cannot be accurately tracked by any reasonable algorithm that makes use of global information. The algorithm we shall construct belongs to this class of algorithms, and we need the value of $n_{ij}^{T}$ in our network model. Therefore, we will estimate $n_{ij}^{T}$ given $n_{ij}$. For ease of analysis, we assume that the durations of talkspurts are exponentially distributed with mean $\phi^{-1}$, and that the durations of silent periods are exponentially distributed with mean $\theta^{-1}$. Then the speaker activity for the off-hook (i,j) conversations can be modelled by the Markov chain shown in figure 1. Using the mean value of $n_{ij}^T$ as its estimate we obtain [16]

$$\overline{n_{ij}^T} = \left(\frac{\theta}{\theta+\phi}\right) n_{ij} = \gamma n_{ij} \tag{1}$$

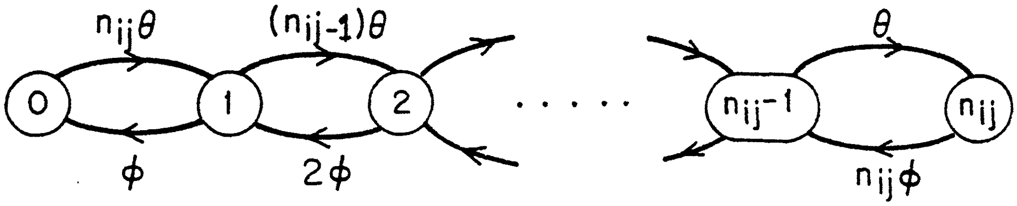


Figure 1. Speaker Activity Model for Off-hook (i,j) Conversations

Figure 2 shows the possible path delays of five packets generated during a talkspurt. These packets are routed along a three-link path $\{(i,k), (k,\ell), (\ell,j)\}$, and they arrive at the destination node at instants indicated on the time axis labelled "Arrival". Generally the intervals between packet arrivals are not uniform. Thus if these packets are delivered to the sink as soon as they arrive at the destination node, the reconstructed speech will contain many uneven and annoying gaps. Therefore, at each destination node smoothing buffers are installed in which the arriving voice packets are temporarily stored and finally released to the sink at the same rate at which the packets entered the network. The time axis labelled "Delivery" in figure 2 shows when these packets are delivered to the sink. Any packet that arrives later than it is required for delivery to the sink is discarded. For example, packet #4 is discarded because $d_3 > \tau_3$. A fictitions packet could be sent to the sink when the real packet arrives late (or alternatively we could repeat the previous packet). Note that fixing the time $\tau_0$ at which the first packet is delivered to the sink sets an upper limit to the time a subsequent packet is expected to arrive at the smoothing buffers. Thus the reason for the extra delay $\tau_0 - d_0$ imparted on the first packet is to help minimize the likelihood of late arrival of subsequent packets. The requirement that voice packets be delivered to the sink almost in real time demands that $\tau_0 - d_o$ cannot be made very large.

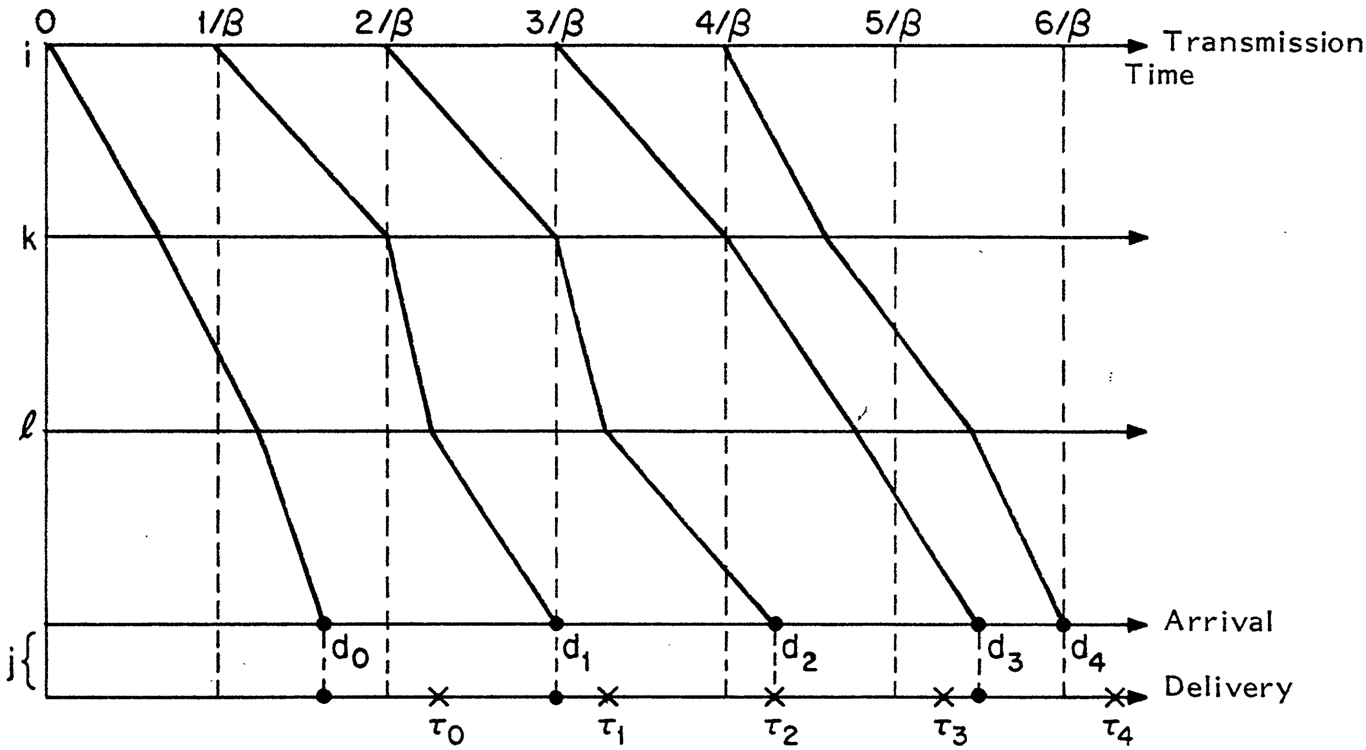


Figure 2. Voice Packet Manipulation at Destination Node

Two main objectives in the above scheme are to reduce the path delay of each voice packet, and to ensure that the quality of the conversation is high. However, high quality is associated with long packets which in turn generate high traffic levels with the attendant larger packet delays. Thus the realization of these objectives resides in making tradeoffs. This then suggests that we formulate an optimization problem whose solution will yield the optimal voice packet length (and hence speech quality) for each network condition.

In order to minimize the dependence of the fate of subsequent packets on $\tau_0$ (and hence $d_0$), we attempt to make all voice packets experience identical path delays on the same route through imposing some cost on packet path delays. Specifically, we consider a cost to be associated with the link delay of each voice packet and each data packet. Since the delay requirements of voice packets are more stringent than those of data packets, we grant the voice packets a non-preemptive priority over the data packets.

Let $W^v_{ik}$ = the expected waiting time of a voice packet at link (i,k)

$W^d_{ik}$ = the expected waiting time of a data packet at link (i,k)

$$W = (W^v_{ik}, W^d_{ik})^T$$

$$F = (F^v_{ik}, F^d_{ik})^T$$

where $(\ )^T$ denotes the transpose of the matrix.

Then $W = f(F)$. That is, the expected waiting times are functions of the aggregate voice and data traffics. For example, if we assume that the queue at each link is an $M|G|1$ queue with priority, then it is well known [17] that the expected waiting times of voice and data packets at each link (i,k) are given, respectively, by

$$W_{ik}^{v} = \frac{W_r}{1 - \rho_v} = \frac{W_r C_{ik}}{C_{ik} - F_{ik}^{v}} \tag{2}$$

$$W_{ik}^{d} = \frac{W_r}{(1 - \rho_v)(1 - \rho)} = \frac{W_r C_{ik}^2}{(C_{ik} - F_{ik}^{v})(C_{ik} - F_{ik})} \tag{3}$$

where $\rho_v$ = the link voice utilization factor

$\rho$ = the link total utilization factor

$W_r$ = the expected remaining time to completion of current service

$$= \frac{1}{2}[\lambda_v E(\tau_v^2) + \lambda_d E(\tau_d^2)] \tag{4}$$

where $\lambda_v$ = the rate at which voice packets arrive at the link

$\lambda_d$ = the rate at which data packets arrive at the link

$\tau_v$ = the time to service one voice packet

$\tau_d$ = the time to service one data packet

Because of the priority which voice packets have over data packets, the effect of data traffic on the expected waiting time of a voice packet is rather small. Hence we define a congestion cost function $B_{ik}(F_{ik}^{v})$ to be the cost of limiting voice traffic on link (i,k) to $F_{ik}^{v}$. We assume that $B_{ik}(F_{ik}^{v})$ is a convex increasing and twice differentiable function of $F_{ik}^{v}$ with the typical plot as shown in figure 3. The reason for the cutoff at $\sigma_1 C_{ik}$, for $0 < \sigma_1 < 1$, is because, from (2) for example, the expected waiting time approaches infinity as $F_{ik}^{v}$ approaches $C_{ik}$. Thus limiting the average voice traffic on each link to some fraction $\sigma_1$ of the link capacity maintains a tolerable expected waiting time on the link.

One major attraction of the integrated packet network is the anticipated possibility of exploiting the on-off characteristics of voice traffic to transmit more data when voice traffic is low. In particular, when the voice traffic $F^v_{ik}$ is zero, data packets should use link (i,k) as in an all-data network. Since data packet delay is not as critical as voice packet delay, we permit a higher cutoff point, say $\sigma_2 C_{ik}$, for data flow on each link (i,k). A cutoff is necessary to account for the finite buffer spaces available at each link. We then define another congestion cost function $G_{ik}(F_{ik})$ to be the cost of limiting the total traffic on link (i,k) to $F_{ik}$. We assume that $G_{ik}(F_{ik})$ is a convex increasing and twice differentiable function of $F_{ik}$, with the typical plot as shown in figure 3.

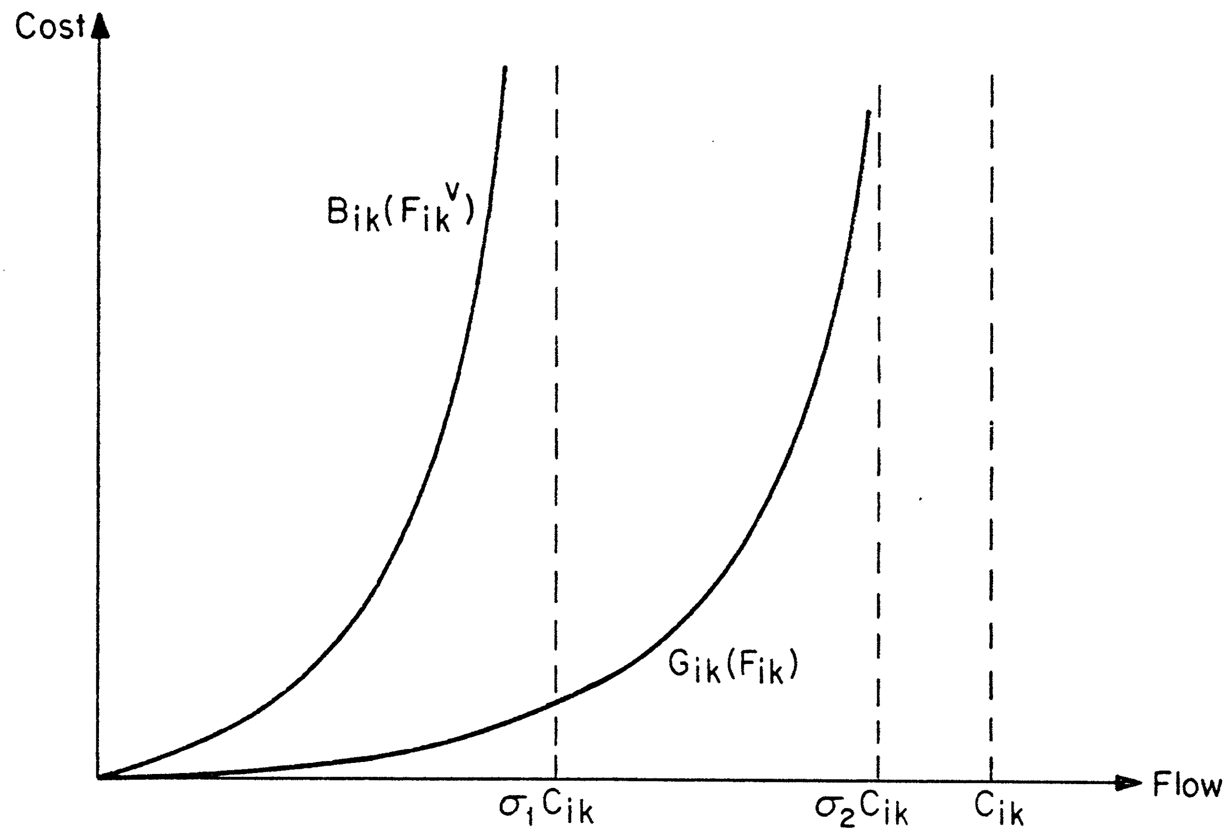


Figure 3. Congestion Cost Functions

Define the composite cost function $D_{ik}(F_{ik}, F_{ik}^{v})$ as follows:

$$D_{ik}(F_{ik}, F_{ik}^{v}) = B_{ik}(F_{ik}^{v}) + G_{ik}(F_{ik}) \tag{5}$$

This is a generalization of the type of congestion cost function which Gallager and Golestaani [18] defined for a data network.

Each user usually derives some satisfaction from transmitting a high quality conversation; and the higher the quality, the more satisfied he is. Thus we associate a cost with the quality of each voice conversation. We define a cost function $V_{ij}(\ell_{ij})$ to be the cost of restricting the packet length of an (i,j) voice conversation to $\ell_{ij}$. We assume that $V_{ij}(\ell_{ij})$ is a convex non-increasing and twice differentiable function of $\ell_{ij}$, with a typical plot as shown in figure 4. There are two reasons for choosing this type of function. The first reason is ease of analysis. The second reason is this: We note that $V'_{ij}(\ell_{ij})$ increases monotonically (from minus infinity toward zero), which implies that it is costly to degrade the quality of any voice conversation by assigning shorter packets to that conversation. This represents a way to ensure fairness to all users. We define $\alpha$ to be the maximum length of any voice packet. The value $\alpha$ accounts for the fact that there is a limit to the number of bits the digitizer can generate in $1/\beta$ seconds. It also accounts for the fact that no appreciable improvement in voice quality is achieved when the packet length exceeds a certain value. If $\alpha_1$ bits is the number of bits the digitizer can generate in $1/\beta$ seconds and $\beta\alpha_2$ bits/sec is the threshold rate for high quality speech, then $\alpha \leq \min[\alpha_1, \alpha_2]$. For notational simplicity we assume that all (i,j) voice conversions have the same $V_{ij}(\ell_{ij})$.

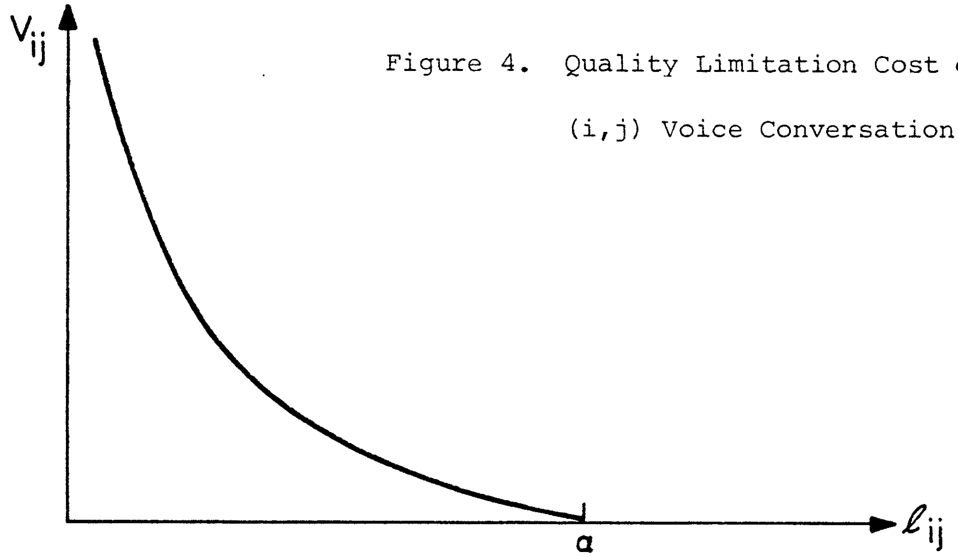


Figure 4. Quality Limitation Cost of an (i,j) Voice Conversation

We assume that if no control was imposed on it an (i,j) data source would transmit at a desired average rate $r_{ij}^{d}$. When control is imposed the user will be more satisfied the closer to $r_{ij}^{d}$ the assigned rate $r_{ij}$ is. Thus we associate with each (i,j) data session a cost function $E_{ij}(r_{ij})$, which is the cost of restricting the average input rate of the session to $r_{ij}$. We assume that $E_{ij}(r_{ij})$ is a convex non-increasing and twice differentiable function of $r_{ij}$. As in the case of voice conversations, the rate limitation cost function $E_{ij}(r_{ij})$ is also used to ensure that all data sessions are fairly treated. For notational simplicity we assume that all (i,j) data sessions have the same $E_{ij}(r_{ij})$. A typical plot of $E_{ij}(r_{ij})$ is shown in figure 5.

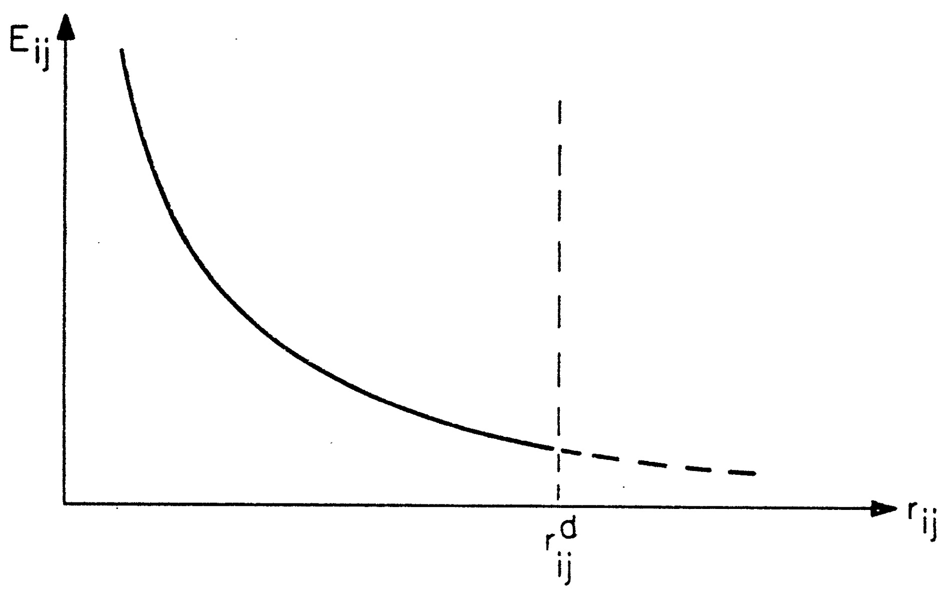


Figure 5. Rate Limitation Cost of an (i,j) Data Session

Let $B_T = \sum_{i,k} B_{ik}(F^v_{ik})$

$$G_T = \sum_{i,k} G_{ik}(F_{ik})$$

$$V_T = \sum_{i,j} \gamma n_{ij} V_{ij}(\ell_{ij})$$

$$E_T = \sum_{i,j} m_{ij} E_{ij}(r_{ij})$$

where $\gamma n_{ij}$ is, as stated earlier, our estimate of $n^T_{ij}$. Then the integrated network flow problem can be formulated as the following nonlinear optimization problem:

$$\text{Minimize } J = B_T + G_T + V_T + E_T \tag{6}$$

$$\text{s.t.} \sum_{k\in O(i)} f^v_{ik}(j) - \sum_{m\in I(i)} f^v_{mi}(j) = \beta\gamma n_{ij}\ell_{ij},\ 1 \leq i,j \leq N \tag{7}$$

$$\sum_{k\in O(i)} f^d_{ik}(j) - \sum_{\ell\in I(i)} f^d_{\ell i}(j) = m_{ij}r_{ij},\ 1 \leq i,j \leq N \tag{8}$$

$$\left.\begin{array}{l} f^v_{ik}(j) \geq 0 \\ f^d_{ik}(j) \geq 0 \end{array}\right\} \quad 1 \leq i,j,k \leq N;\ i \neq j \tag{9}$$

Constraints (7) and (8) are the so-called continuity (or flow conservation) equations. They state that the total average traffic coming into node i and destined for node j is equal to the total average traffic going out of node i and destined for node j. Constraint (9) states that all flows are non-negative.

## III. A Joint Flow Control and Routing Algorithm

Theorem 1 Let $u^* = [\ell^*_{ij}, r^*_{ij}, f^{v*}_{ik}(j), f^{d*}_{ik}(j)]$ be a feasible point of (7) through (9). Then $u^*$ minimizes (6) if and only if there exist two sets of numbers $\lambda = \{\lambda_{ij}\}$ and $\mu = \{\mu_{ij}\}$, with $\lambda_{jj} = 0$ and $\mu_{jj} = 0$, such that the following Kuhn-Tucker conditions are satisifed [13], [14]:

$$B'_{ik}(F^{v*}_{ik}) + G'_{ik}(F^{*}_{ik}) + \lambda_{kj} \begin{cases} = \lambda_{ij} & \text{if } f^{v*}_{ik}(j) > 0 \\ \geq \lambda_{ij} & \text{if } f^{v*}_{ik}(j) = 0,\ i \neq j \end{cases} \tag{10}$$

$$G'_{ik}(F^{*}_{ik}) + \mu_{kj} \begin{cases} = \mu_{ij} & \text{if } f^{d*}_{ik}(j) > 0 \\ \geq \mu_{ij} & \text{if } f^{d*}_{ik}(j) = 0,\ i \neq j \end{cases} \tag{11}$$

$$p_{ij}(\ell^*_{ij}) \begin{cases} = \lambda_{ij} & \text{if } 0 < \ell^*_{ij} < \alpha \\ \leq \lambda_{ij} & \text{if } \ell^*_{ij} = 0 \\ \geq \lambda_{ij} & \text{if } \ell^*_{ij} = \alpha \end{cases} \tag{12}$$

$$q_{ij}(r^*_{ij}) \begin{cases} = \mu_{ij} & \text{if } 0 < r^*_{ij} < r^d_{ij} \\ \leq \mu_{ij} & \text{if } r^*_{ij} = 0 \\ \geq \mu_{ij} & \text{if } r^*_{ij} = r^d_{ij} \end{cases} \tag{13}$$

where $B'_{ik}(F^v_{ik})$ is the derivative of $B_{ik}(F^v_{ik})$, etc; and $p_{ij}(\ell_{ij}) = -\frac{1}{\beta} V_{ij}(\ell_{ij})$ and $q_{ij}(r_{ij}) = -E'_{ij}(r_{ij})$ are called the voice and data priority functions, respectively.

We define a voice (data) route $R^v_{ij}$ $(R^d_{ij})$ between nodes i and j as the set of links $\{(i,k), (k,\ell),\ldots,(m,j)\}$ along which (i,j) voice (data) conversations flow. We can interpret $G'_{ik}(F_{ik})$ as the marginal cost of data traffic on link (i,k) or the "data length" of link (i,k). Similarly, we can interpret $B'_{ik}(F^v_{ik}) + G'_{ik}(F_{ik})$ as the marginal cost of voice traffic on link (i,k) or the "voice length" of link (i,k). Since $\lambda_{jj} = \mu_{jj} = 0$, we can solve for $\lambda_{ij}$ and $\mu_{ij}$ recursively to obtain

$$\lambda_{ij} = \sum_{(\ell,k)\in R^v_{ij}} \{B'_{\ell k}(F^{v*}_{\ell k}) + G'_{\ell k}(F^*_{\ell k})\} \tag{14}$$

$$\mu_{ij} = \sum_{(\ell,k)\in R^d_{ij}} G'_{\ell k}(F^*_{\ell k}) \tag{15}$$

Then $\lambda_{ij}$ $(\mu_{ij})$ can be interpreted as the marginal voice (data) cost of congestion on a path $R^v_{ij}$ $(R^d_{ij})$ of optimal flow, or the voice (data) length of path $R^v_{ij}(R^d_{ij})$ when flow is optimal. Equations (10) and (11) then state that all traffic flows on paths of minimum marginal cost, i.e. the shortest paths. In (14) and (15) $\lambda_{ij}$ and $\mu_{ij}$ are defined as properties of the optimal flow and are difficult to find without solving the optimization problem. We redefine $\lambda_{ij}$ and $\mu_{ij}$ in terms of measurable quantities, as functions of an arbitrary flow:

$$\lambda_{ij} = \min_{R^v_{ij}} \sum_{(\ell,k)\in R^v_{ij}} \{B'_{\ell k}(F^v_{\ell k}) + G'_{\ell k}(F_{\ell k})\} \tag{16}$$

$$\mu_{ij} = \min_{R^d_{ij}} \sum_{(\ell,k)\in R^d_{ij}} G'_{\ell k}(F_{\ell k}) \tag{17}$$

Then when flow is optimal the $\lambda_{ij}$ obtained from (14) is equal to that obtained from (16), and similarly for $\mu_{ij}$.

The voice priority functions $p_{ij}(\ell_{ij})$ are the marginal gain in voice quality for an additional voice packet length allocation. Equations (12) then states that optimality occurs when the marginal gain in voice quality is as close as possible to the marginal voice cost of congestion, $\lambda_{ij}$, subject to $0 < \ell^*_{ij} < \alpha$. The point of optimal tradeoff is attained when these two marginal values are equal. The same explanation holds for data sessions. A desirable flow control algorithm would then be one that attempts to equalize these two sets of marginal values for each conversation.

From (10) through (13) we observe that the voice and data distances $\lambda_{ij}$ and $\mu_{ij}$ are the main network parameters required to find the routes, and to set the voice packet lengths and the data input rates. Thus any quasi-static routing algorithm, such as [15], [19] and [20], can be modified to jointly perform the functions of routing and flow control. There are several ways to construct the joint flow control and routing algorithm, but we will define a simple one in the space of path flows.

Let $S^{vn}_{ij} = \beta\gamma n_{ij}\, \ell^n_{ij}$

= the total expected rate of all the (i,j) voice conversations at iteration n.

$S^{dn}_{ij} = m_{ij}\, r^n_{ij}$

= the total expected rate of all the (i,j) data conversations at iteration n

$A_v = \{(i,j) \mid S^{vn}_{ij} > 0;\ i,j = 1,\ldots,N\}$

= the set of voice conversations in the network

$$A_d = \{(i,j) \mid S_{ij}^{dn} > 0;\ i,j = 1,\ldots,N\}$$

= the set of data sessions in the network

$P(i,j)$ = the set of directed paths, with no repeated nodes, originating at node i and terminating at node j

$S_p^{vn}$ = the total expected voice traffic from the (i,j) voice conversations routed along path $p \in P(i,j)$ at iteration n

$S_p^{dn}$ = the total expected data traffic from the (i,j) data sessions routed along path $p \in P(i,j)$ at iteration n.

Then the relationship between the link flows and the path flows at iteration n is given by

$$F_{ik}^{vn} = \sum_{(i,j)\in A_v} \sum_{p\in P(i,j)} \delta_p(i,k)\ S_p^{vn} \tag{18}$$

$$F_{ik}^{dn} = \sum_{(i,j)\in A_d} \sum_{p\in P(i,j)} \delta_p(i,k)\ S_p^{dn} \tag{19}$$

where the incidence term $\delta_p(i,k)$ is defined as follows:

$$\delta_p(i,k) = \begin{cases} 1 & \text{if link } (i,k)\in p \\ 0 & \text{otherwise} \end{cases}$$

Let $\Lambda_{ik}^n = B'_{ik}(F_{ik}^{vn}) + G'_{ik}(F_{ik}^n)$

= the voice length of link (i,k) at iteration n

$\Pi_{ik}^n = G'_{ik}(F_{ik}^n)$

= the data length of link (i,k) at iteration n

Then for each (i,j) voice conversation and path $p \in P(i,j)$ we define the voice length of path p at iteration n as

$$\Lambda_p^n = \sum_{(\ell,k)\in p} \Lambda_{\ell k}^n \tag{20}$$

Similarly, for each (i,j) data conversation and path $p \in P(i,j)$ we define the data length of path p at iteration n as

$$\Pi_p^n = \sum_{(\ell,k)\in p} \Pi_{\ell k}^n \tag{21}$$

Then the shortest voice and data distances for the (i,j) conversations at iteration n are given, respectively, as

$$\lambda_{ij}^n = \underset{p\in P(i,j)}{\text{Min}} \Lambda_p^n \tag{22}$$

$$\mu_{ij}^n = \underset{p\in P(i,j)}{\text{Min}} \Pi_p^n \tag{23}$$

We formally define the join flow control and routing algorithm as follows:

A. <u>Update at Iteration n</u>

1. Each node i broadcasts $B'_{ik}(F_{ik}^{vn})$ and $G'_{ik}(F_{ik}^n)$ for each $k \in O(i)$.

2. After receiving the above information from all nodes, node i computes the voice path length $\Lambda_p^n$ for each $p \in P(i,j)$ with $S_p^{vn} > 0$, and the data path length $\Pi_p^n$ for each $p \in P(i,j)$ with $S_p^{dn} > 0$ according to (20) and (21) respectively.

3. Node i determines the minimum distances $\lambda_{ij}^{n}$ and $\mu_{ij}^{n}$ for each j such that $(i,j) \in A_v$, and each j such that $(i,j) \in A_d$ according to (22) and (23) respectively.

4. Node i computes the voice priority function $p_{ij}(\ell_{ij}^{n})$ for each $(i,j) \in A_v$, and the data priority function $q_{ij}(r_{ij}^{n})$ for each $(i,j) \in A_d$.

B. <u>Voice Flow Control</u>

For each j such that $(i,j) \in A_v$, node i does the following:

1. If $\lambda_{ij}^{n} > p_{ij}(\ell_{ij}^{n})$, decrease $\ell_{ij}^{n}$ by an amount proportional to $\lambda_{ij}^{n} - p_{ij}(\ell_{ij}^{n})$, if $\ell_{ij}^{n} > 0$.

2. If $\lambda_{ij}^{n} < p_{ij}(\ell_{ij}^{n})$, increase $\ell_{ij}^{n}$ by an amount proportional to $p_{ij}(\ell_{ij}^{n}) - \lambda_{ij}^{n}$, if $\ell_{ij}^{n} < \alpha$.

3. If $\lambda_{ij}^{n} = p_{ij}(\ell_{ij}^{n})$, leave $\ell_{ij}^{n}$ unchanged.

C. <u>Data Flow Control</u> Similar to B above.

D. <u>Voice Traffic Routing</u>

1. For each $p \in P(i,j)$ that is not a shortest path node i decreases the traffic by an amount proportional to $\Lambda_{ij}^{n} - \lambda_{ij}^{n}$.

2. Node i puts the remaining (i,j) traffic on the shortest path. If two or more shortest paths exist, node i arbitrarily chooses one of them to perform the functions of the shortest path.

E. Data Traffic Routing

Similar to D.

It is shown in [14] that if the constants of proportionality are sufficiently small, the joint flow control and routing alogrithm converges to the optimal solution. Note that we compute the voice path lengths of those paths with $S_p^{vn} > 0$ and the data path lengths of those paths with $S_p^{dn} > 0$. This is because it is usually not computationally feasible to find the path length of every path from node i to node j. A certain amount of synchronization of link flow broadcasts is necessary in the joint flow control and routing algorithm.

## IV. A Congestion Control Protocol

In this paper we distinguish between flow control and congestion control, even though the two terms are often used interchangeably in the literature. As defined earlier, by flow control we mean the set of mechanisms used to regulate the entry of traffic into the network. Thus the setting of voice packet lengths and data input rates to match the network conditions is a flow control mechanism. By congestion control we shall mean the set of mechanisms used to reduce high network loading by manipulating the traffic already admitted into the network. Thus by this definition, the embedded coding scheme [11] is a congestion control mechanism.

Despite the flow control we practice, network congestion can still take place. This can be caused by several factors including the fact that our algorithm is designed for networks with slowly varying input statistics while in practice the integrated network may have rapidly varying input statistics. In order to contain occasional incidence of network congestion, we propose to practice congestion control in conjunction with the proposed voice flow control. Our congestion control protocol is a modification of the embedded coding scheme [11]. We assume that after the packet lengths have been set, each packet is then encoded into different segments of different orders of importance. As in [11], the less important segments may be discarded anywhere in the network when congestion sets in. The more important segments reaching the destination are considered capable of producing usable but lower quality speech. In this scheme, a voice sink has no need to report its rate of packet acceptance to the source node, as is done in [11], because the joint flow control and routing algorithm will correct any rate mismatch at the next iteration as part of its flow control function.

Consider a somewhat practical way to implement the voice flow control and congestion control protocols of our scheme. Assume that the voice digitizer emits packets of fixed length $\alpha$ bits every $1/\beta$ seconds, after encoding them into different segments of different orders of importance. The source node, which knows what the appropriate packet lengths should be at each iteration, then strips some less important segments of the packet to generate the right packet lengths before sending these packets along their paths. Finally, any node may discard more of the less important segments if congestion builds up in the network.

## V. Discussion

The work reported in [21] can be considered as a generalization of the incremental delay routing principles of Gallager [15], and in this paper we have extended the idea in [21] to integrated networks. However, we have concentrated on the issue of voice packet delay at the expense of some other details. For example, we have left out the details of the dynamic implementation of data flow control. Also even though it is difficult to use the set of instantaneous values of $n_{ij}^{T}$ on a global basis for flow control, it is possible for each source node i to use the instantaneous values of $n_{ij}^{T}$ as a guide to avoiding congestion. Finally, it is possible for the smoothing buffers to deliver voice packets to the sink at a rate smaller than $\beta$, especially when congestion builds up and voice packets start to arrive at the smoothing buffers slower than these packets are delivered to the sink. Details of these issues can be found in [14].

References

[1] I. Gitman and H. Frank, "Economic Analysis of Integrated Voice and Data Networks: A Case Study," Proc. IEEE, vol. 66, No. 11, pp. 1549-1570, November 1978.

[2] J. W. Forgie and A. G. Nemeth, "An Efficient Packetized Voice/Data Network Using Statistical Flow Control," ICC Conference Records, vol. III, pp. 44-48, June 1977.

[3] G. Coviello and P. Vena, "Integration of Circuit/Packet Switching in a SENET (slotted envelope network) Concept," NTC Conference Records, pp. 42-12 to 42-17, December 1975.

[4] P. Baran, "On Distributed Communication Networks," IEEE Trans. Comm. Systems, vol. CS-13, pp. 1-9, March 1964.

[5] L. Kleinrock, Communication Nets: Stochastic Message Delay, New York: McGraw-Hill Book Co., 1964.

[6] B. D. Wessler and R. B. Hover, "Public Packet-switched Networks," Datamation, vol. 20, No. 7, pp. 85-87, July 1974.

[7] L. Tymnes, "TYMNET - A Terminal-Oriented Communication Network," AFIPS Conf. Proc., vol. 38, pp. 211-216, 1971.

[8] A. Daret et al., "The French Public Packet Switching: The Transpac Network," Proc. 3rd ICCC, pp. 251-260, Toronto, 1976.

[9] W. W. Clipsham et al., "Datapac Network Overview," Proc. 3rd ICCC, pp. 131-136, Tronoto, 1976.

[10] M. Gerla and L. Kleinrock, "Flow Control: A Comparative Survey," IEEE Trans. Comm., vol. COM-28, No. 4, pp. 553-574, April 1980.

[11] T. Bially, B. Gold, and S. Seneff, "A Technique for Adaptive Voice Flow Control in Integrated Packet Networks," IEEE Trans. Comm., vol. COM-28, No. 3, pp. 325-334, March 1980.

[12] S. Seneff, "Computer Simulation Model for a Digital Communication Network Utilizing an Embedded Coding Technique," MIT Lincoln Lab., Lexington, MA, Tech Note 1978-33, October 1978.

[13] D. G. Luenberger, Introduction to Linear and Nonlinear Programming, Reading, MA: Addison-Wesley Publishing Company, 1973.

[14] O. C. Ibe, "Flow Control and Routing in an Integrated Voice and Data Communication Network," Report LIDS-TH-1115, August 1981, Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139.

[15] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," IEEE Trans. Comm., vol. COM-25, No. 1, pp. 73-85, January 1977.

[16] L. Kleinrock, Queueing Systems, vol. 1, New York: John Wiley and Sons, Inc., 1975.

[17] L. Kleinrock, Queueing Systems, vol. 2, New York: John Wiley and Sons, Inc., 1976.

[18] R. G. Gallager and S. J. Golestaani, "Flow Control and Routing Algorithms for Data Networks," Proc. 5th ICCC, Atlanta, GA, October 1980.

[19] D. P. Bertsekas, "A Class of Optimal Routing Algorithms for Communication Networks," Proc. 5th ICCC, Atlanta, GA, October 1980.

[20] A. Segall, "Optimal Distributed Routing for Virtual Line-Switched Data Networks," IEEE Trans. Comm., vol, COM-27, No. 1, pp. 201-209, January 1979.

[21] S. J. Golestaani, "A Unified Theory of Flow Control and Routing in Data Communications Networks," Report LIDS-TH-963, January 1980, Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139.