

SNA*/SDLC PERFORMANCE OVER ISDN FRAME-RELAY, VIRTUAL-CIRCUIT DATA NETWORKS

Anurag Kumar

Anurag Kumar is a member of technical staff in the Performance Analysis Department at AT&T Bell Laboratories in Holmdel, New Jersey. He has been involved in analysis of electronic switching systems, data communication networks, and survivable networks. Mr. Kumar joined the company in 1981 and has a B.S. from the Indian Institute of Technology, Kanpur, India, and a Ph.D. from Cornell University, both in electrical engineering.

Data transport over permanent virtual circuits (PVCs) in a high-speed, packet-switching data network can be a cost-effective alternative to using long-distance, voice-grade, or Dataphone® digital service private-line networks. In this paper, we analyze the performance of several architectures for providing SNA/SDLC services over PVCs in an ISDN frame-relay data network. In the simplest architecture, a PVC replaces a network private line, and short private lines (still under SNA/SDLC protocol) are used to access the network. Several alternative configurations (e.g., remote or local polling) are possible, requiring various degrees of sophistication. We have analyzed their delay performance using several queueing models, and compared it to the private-line case. Our results have helped identify those combinations that yield the best performance. For example, pipelining is essential for good delay performance in a simple replacement of a private line with a PVC, and local polling has significant performance advantages when several low-speed, multidrop "tail" circuits are multiplexed onto a high-speed front-end processor access line.

Introduction

With the advent of high-speed, virtual-circuit data networks, data transport over permanent virtual circuits can be a cost-effective alternative to data transport over long-distance, voice-grade, or Dataphone® digital-service private lines. The latter provide digital data service (DDS). (Panel 1 defines acronyms used in this paper.) A virtual-circuit network solution offers customers superior reliability and the possibility of easy reconfiguration.

* Trademark of IBM Corporation.

Panel 1. Acronyms in This Paper

CC	cluster controller
DDS	digital data service
FEP	front-end processor
FIFO	first-in, first-out
ISDN	Integrated Services Digital Network
LAPD	link-access procedures for D channel
PVC	permanent virtual circuit
SDLC	synchronous data link control
TA	terminal adapter
TA _C	terminal adapter on the CC side
TA _F	terminal adapter on the FEP side

28 Most private lines are used to connect host computers and remote terminals in business environments—such as banking, airline reservations, and point-of-sale businesses—where a terminal user and host computer interact through enquiries and responses. In this paper, we describe several alternative ISDN (Integrated Services Digital Network) frame-relay virtual-circuit network solutions¹ for such an enquiry-response type application. Next, we analyze their performance using queueing models and compare it to that of existing private-line configurations. Our principal concern is the end-to-end data delay performance of the transport mechanisms and the load imposed on data network resources.

The Configurations. Figure 1a shows a typical multipoint, private-line configuration. The host accesses the line through a front-end processor (FEP). The remote terminals are grouped into clusters, and each cluster accesses the private line via a cluster controller (CC). We refer to this configuration as the *baseline case*.

Because the same private line connects several CCs to the FEP, one needs some type of multiple access protocol. In the most common implementation of such systems, the FEP polls the CCs using the link-level procedures of IBM's SNA™ systems-network protocol—namely, SDLC (synchronous data link control) in the normal response mode.

Simple replacement. In this simple replacement of a private line with a virtual circuit (Figure 1b), we still use short-distance private lines (with the SNA/SDLC protocol)

to access the network; terminal adapters (TAs) perform protocol encapsulation or conversion. (TAs provide the interface between the native protocol and the access protocol of the virtual-circuit network.) In Figure 1b, TA_F is the terminal adapter on the FEP side, TA_C is the terminal adapter on the CC side, and a virtual circuit over several network trunks and switches connects the two adapters. We shall refer to this configuration as *simple private-line replacement*.

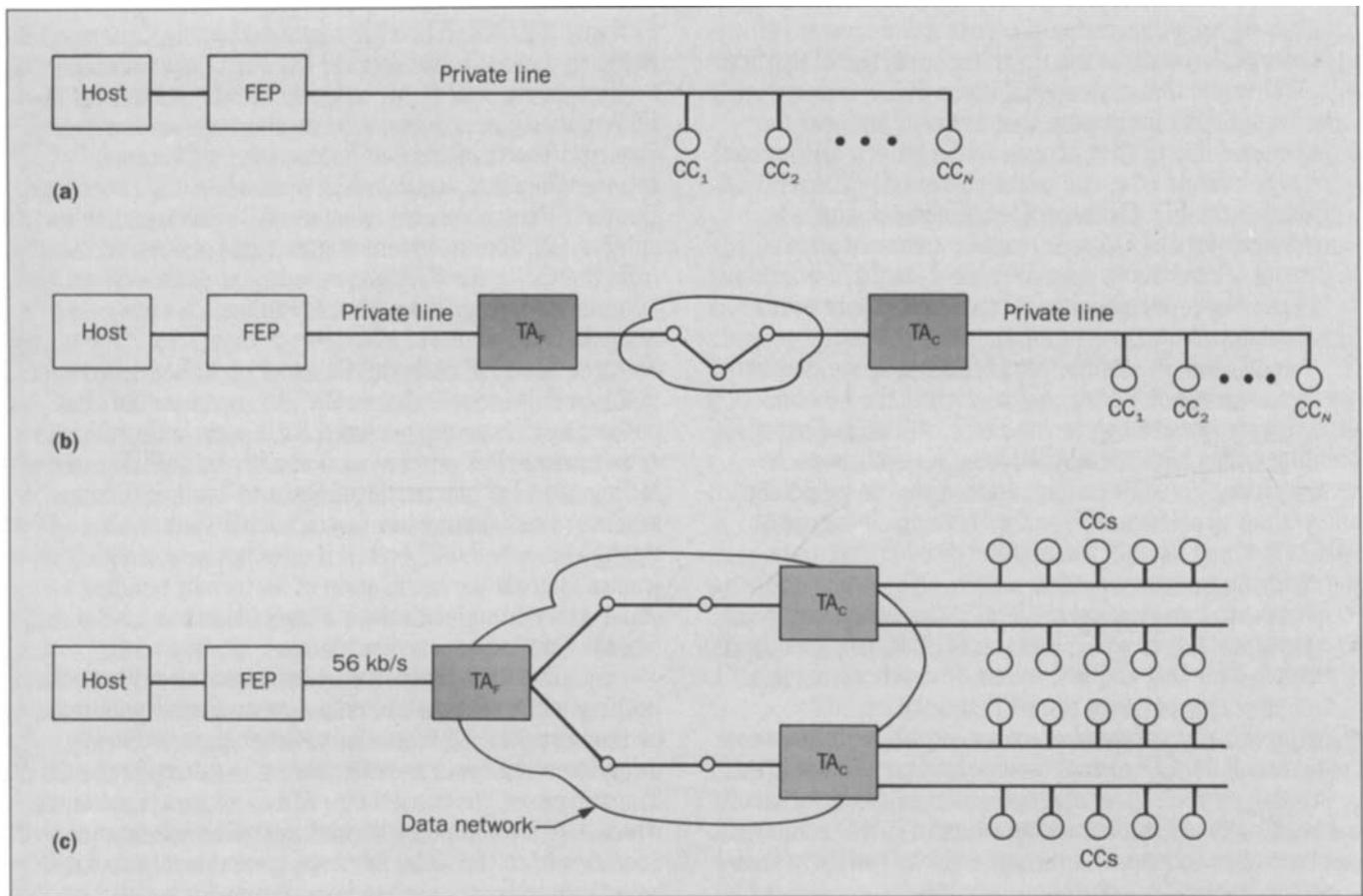
Logical multipoint. With a virtual-circuit network available to replace private lines, several more sophisticated—and perhaps better performing—architectural alternatives become available. One alternative is to use the network to multiplex several low-speed, multipoint “tail” circuits (short private lines on the CC side) onto one high-speed line [e.g., a 56-kilobits-per-second (kb/s) DDS line] that terminates at a high-speed port on a FEP in the data center (Figure 1c). Because of software restrictions, only one poll can be outstanding on each FEP port. But if we multiplex several multipoint circuits into one high-speed FEP port, the FEP “sees” them as a single multipoint line; hence, the term *logical multipoint* for this configuration.

Virtual-circuit options. The virtual-circuit-network solution affords another possibility called *fanout*. Here, each of the existing multidrop lines, with its several CCs, can be separated into several multidrop lines, each with its own TA port and, of course, fewer CCs. The fanout can be *partial* or *full*. The latter means that a multidrop line with, say, N CCs is separated into N point-to-point lines, each with just one CC.

We shall assume that LAPD (link-access procedures for D channel) is used for data transfer in the network. TAs wrap SDLC frames into LAPD frames, and network switches use frame-relay procedures.

If virtual circuits in a data network (and the requisite protocol adapters) replace the private lines, then we have several implementation options:

- **Remote polling**—Every SDLC frame (including information frames, poll frames, and other supervisory frames) is “wrapped” into the network protocol and transported transparently across the network. The FEP thus polls the CCs across the network; hence the term, remote polling.



- Local Polling**—An adapter, TA_C, on the CC side polls the CCs, accepts enquiries from them, and ships these enquiries across the network to the adapter, TA_F, on the FEP side. TA_F queues these enquiries and behaves like several “virtual” CCs (one for each real CC). The FEP polls TA_F for enquiries and returns responses to TA_F, which then ships them to the appropriate TA_C. The TA_Cs transmit the responses to the CCs over the multipoint private lines on the remote side.
- Pipelining**—To mitigate the additional store-and-forward delay, TAs can form LAPD frames and ship them over the network before an entire SDLC frame has arrived at the network edge. This is called pipelining into the net-

Figure 1. Network configurations analyzed. (a) Private line configuration (baseline case). The host computer accesses the private line via a front-end processor (FEP). A cluster controller (CC) provides network access for each cluster of remote terminals. (b) Simple private line replacement. A virtual circuit replaces a private line and terminal adapters (TAs) provide protocol interfaces. The FEP gives the host computer access to a short private line. (c) Multiplexing several multipoint tail circuits onto a high-speed FEP line.

work. At the other end of the virtual circuit, a small "build-out" delay is enforced after the arrival of the first LAPD frame that corresponds to an SDLC frame. Newly arriving LAPD frames are then transmitted over the low-speed line as they arrive, until a bit in a TA protocol header indicates the end of the current SDLC frame. Because the SDLC protocol is bit-oriented and synchronous, an SDLC frame must be transmitted in a stream of continuous bits. We select the build-out delay to keep the probability of synchronization loss within acceptable limits.

The Analysis. In the rest of this paper, we develop queueing models for three configurations: the baseline case, simple private-line replacement, and logical multipointing with a high-speed FEP line. We analyze each virtual-circuit-network configuration with remote polling and with local polling, with and without pipelining, and with or without fanout. We are interested in these performance measures:

- *Round-trip transport delay*—This is the time from when an enquiry enters a CC until the CC fully receives the response for that enquiry, minus host-processing time and any delay between the FEP and the host.
- *Network traffic overhead*—In remote polling, the network carries all the SDLC supervisory frames. This can result in much more extra traffic in the network than local polling. Expressed as the number of extra LAPD frames carried per second, this measure shows the extra load on the network's protocol processors. Expressed as the number of extra bytes carried per second, this measure shows the extra load on the network's transmission facilities.

SNA/SDLC Protocol Implementation

Details of SDLC procedures and those of similar link-level protocols are well known.² In this section, we informally describe the FEP-CC protocol interactions that are important to our analysis.

Baseline Case. Here, we describe the protocol when a single FEP port and the corresponding CCs communicate over a voice-grade, multipoint private line (i.e., the baseline case).

A physical *full-duplex circuit* exists between the

FEP and the CCs. The FEP can send to a CC whenever it wants to, but a CC can send to the FEP only when the FEP explicitly polls it. We refer to the direction from the FEP to the CCs as *outbound*, and the reverse direction as *inbound*. The CCs are *half-duplex*; i.e., a CC cannot receive while it is sending, nor send while it is receiving. But the FEP can receive from one CC while sending to another CC. In the current context, information frames from the CC to the FEP are called *enquiry frames*, and information frames from the FEP to the CCs are called *response frames*.

The FEP polls the CCs in a cyclic sequence specified by a fixed *service-order table*. At any time, only one poll can be outstanding on each FEP port and carries the sequence number of the next frame that the FEP expects. If the polled CC has enquiry frames to send, it can send, at most, a certain number (equal to the window size) of them. The polled CC, even if it does not send enquiry frames, signals the completion of its turn by sending a supervisory frame with the F (final) bit set to 1. (We shall refer to this frame as a *final frame*.)

The FEP delivers enquiries from the CCs to the host computer and queues response messages generated by the host. The FEP also serves this queue cyclically, using the same service-order table it uses to poll the CCs. The responses are framed into SDLC information frames, which may include piggy-backed acknowledgments to enquiries from the CCs. At most, a window of unacknowledged response frames can exist between the FEP and a CC.

Of course, while the FEP is sending out response frames, it must continue to send polls. Suppose a CC finishes taking its turn, indicates this by sending a final frame, and this final frame arrives in the middle of an outbound transmission from the FEP. Then, the FEP will poll the next CC *after* it finishes sending the current SDLC frame.

Owing to the CC's half-duplex nature, two special situations arise and are handled in this implementation. If the FEP is sending response frames to a CC and it is that CC's turn to be polled, then polling is suspended until all response frames have been transmitted to the CC; after this, the CC is polled. If a CC has been polled and its turn

to receive response frames comes, then that CC loses its turn and the next CC in the service-order table is selected for response-frame transmission.

Remote Polling. In remote polling configurations, the full-duplex circuit between the FEP port and the corresponding CCs consists of a virtual circuit, the private access lines, and the TAs, which may or may not do pipelining. The protocol interactions between the FEP and the CCs are the same as those just described.

Local Polling. In local polling configurations, the TA_F appears as several virtual CCs to each FEP port. Thus, over the full-duplex private access line, the protocol interactions between the FEP and the virtual CCs in a TA_F are as described above. Each tail circuit consists of a full-duplex, voice-grade private line, and the TA_C appears as a virtual FEP to the CCs on that tail circuit. We assume that a TA_C emulates a FEP in terms of its protocol interactions with the CCs.

Basic Issues and Performance Tradeoffs

Even without a quantitative analysis, some basic issues and performance tradeoffs are evident from our description of the system architectures and communication protocols. The main factors that affect performance are:

- Propagation, network, and modem delay and modem switching time
- Queueing and polling of frames
- Pipelining, high-speed line, and fanout.

Because these factors arise repeatedly in understanding the quantitative performance comparisons and tradeoffs among the various configurations, we discuss them qualitatively here.

Propagation Delay and Network Delay. These delays play similar roles. In the baseline case (where only propagation delay figures) and in the virtual-circuit-network case with remote polling, these delays increase the round-trip transaction delay. They increase the CC polling cycle and also, of course, add directly to the transaction transport time. With local polling over short private lines, the polling cycles at the TA_F and the CCs are not affected, but these delays simply add directly to the transaction transport time.

Modem Delay and Modem Switching Time. Modem delay is the signal processing delay in the modem. Modem

switching time is the time during which a primary device on a multipoint line equalizes itself to the inbound channel from a newly polled secondary device.

Modem delay affects every transmission through the modem. Thus, it directly affects the polling cycle and the transaction transport time.

Modem switching time, on the other hand, affects the interval between when a CC receives a poll and when it starts to transmit. Thus, modem switching time increases the polling cycle. But on a point-to-point link, there is no need to reequalize the primary modem each time it polls the secondary device. Hence, there is no modem switching time between the FEP and the TA_F , and between the TA_C and a CC in the full fanout case.

Queueing of Response Frames. At least the first LAPD frame that corresponds to a response frame must be accumulated at a TA_C before transmission of the response frame to a CC can begin. If pipelining is not done, then the entire SDLC frame will have to accumulate in the TA_C . If a high-speed FEP line is used, then response frames will queue up in the TA_C owing to the lower speed CC line.

These observations hold for local polling and remote polling. But for remote polling, a poll for a CC may have to wait behind responses for other CCs queued up in the TA_C , thus adversely affecting the CC polling cycle. This effect will be the worst with a FEP line that is much faster than the CC line because almost all queueing of outbound responses will occur in the TA_C . With local polling, response frames bound for a CC may have to wait in the TA_C for an outstanding poll to that CC.

Queueing and Polling of Enquiry Frames. For remote polling with low-speed access lines, the same comments apply as for response frames; however, enquiry frame delays in the TA_F affect the final frame delay, which in turn affects the polling delay. Of course, these enquiry frames will be coming from the same CC as the final frame. A high-speed FEP line will actually reduce this final frame delay (and thus the polling delay).

With local polling, enquiry frames must queue up in the TA_F , to be picked up by local polls from the FEP. Thus, although each local polling cycle is faster, enquiries must queue up twice to be picked up by polls.

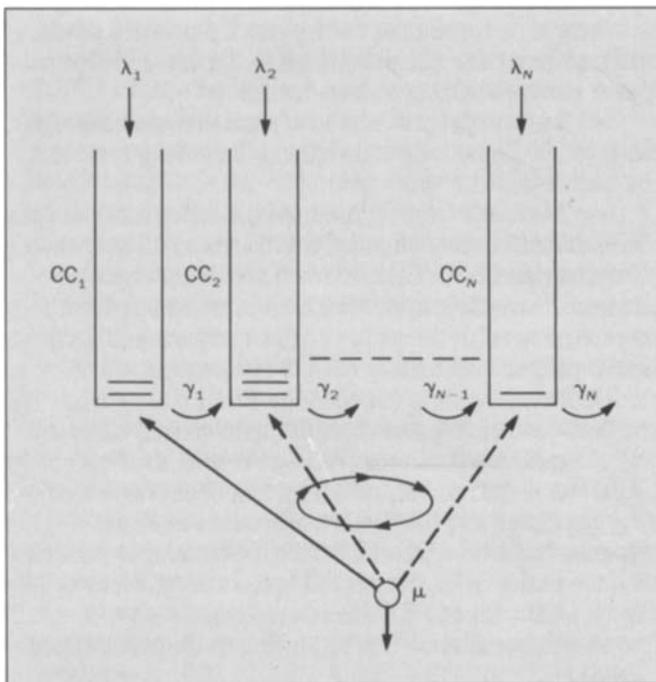


Figure 2. Model for inbound queuing and transmission delays for N cluster controllers (CCs). λ_i is the arrival rate of enquiries to CC_i ; γ_i is the walk-time from CC_i to CC_{i+1} ; μ is the transmission rate.

Pipelining. Owing to the SDLC protocol's synchronous nature, pipelining is only feasible if the downstream access line has a speed equal to or lower than the upstream access line. Thus, when the private access lines are of equal speed, pipelining in both directions is always feasible with remote polling. With a high-speed FEP line, pipelining is never feasible in the inbound direction, and does not yield much savings in the outbound direction.

With equal-speed access links and local polling, pipelining seldom helps in the inbound direction because the enquiries must wait for a poll anyway. Here, pipelining usually helps in the outbound direction, except when a response frame has to wait in the TA_C because the CC to which it is going is being polled.

Overall, pipelining helps remote polling a lot more than it helps local polling.

High-Speed FEP Line. The transmission and queuing time of response frames in the FEP can be greatly reduced with a high-speed FEP line (e.g., 56 kb/s). But if the line between the TA_C and the CCs is low speed, then the net effect is to "move" the response frame queue from the FEP to TA_C . We have already seen the adverse effect of this on remote polling.

The advantage of a high-speed FEP line can be fully achieved only with local polling. When the FEP is able to poll the TA_F rapidly, local polling's double-queuing and double-polling problem is mitigated.

Fanout. Without fanout (i.e., with intact tail circuits), polling and queuing over the low-speed multipoint tail circuit will limit the delay performance to no better than the baseline configuration.

Fanout can benefit remote polling in two ways. First, with fewer CCs on a line, there is a smaller chance that response frames bound for other CCs will delay a poll in a TA_C . Second, with full fanout, there is no modem switching time at the CCs.

Fanout reduces the polling cycle at the CCs, which helps local polling. With full fanout and local polling, each CC is polled independently; an enquiry at a CC will experience the least delay with this configuration. One can expect this implementation, combined with a high-speed FEP line, to yield smaller transport delays than the baseline configuration.

Models for Mean Value Analysis

Here, we describe the queuing models that we have used to analyze the performance of the configurations just described. We use three submodels:

- One for enquiry delays in the CCs and the inbound private line from the CCs
- One for response delays in the FEP and the outbound private line from the FEP
- One for network delays.

With slight modifications, these models also serve to analyze delays in TA_F and TA_C in the local polling configurations. Our principal effort here is to capture the effect of the SDLC polling protocol.

We use a simple network model to capture the average network delays.

Inbound Enquiry Delays Model. The model for inbound enquiry delays is a cyclic server model with server walk-times^{3,4} and is depicted in Figure 2. There are N CCs, namely CC_1, CC_2, \dots, CC_N . Our analysis is for the simplest cyclic service order, i.e., $(CC_1, CC_2, CC_3, \dots, CC_N)$.

Although we could generalize the analysis, we have tied some of our assumptions in the analysis to a particular traffic model that is typical of a class of business applications. In particular, the enquiries are short and elicit responses that are about ten times as long. This implies that outbound transmission will limit the system's transaction carrying capacity.

We assume that each enquiry fits into one SDLC frame, so arrivals of enquiries from the terminals correspond to SDLC frame arrivals. Enquiry frames arrive according to a Poisson process at the rate λ_i to CC_i . The server, which represents the inbound private-line channel, walks from queue to queue, serving at most a window of frames (typically, seven) each time it visits a queue. A poll to a CC corresponds to assigning the server to that CC. The service rate of the server, μ , is just the transmission rate. Hence, if an enquiry frame is l_e bytes long, its service takes l_e/μ seconds.

In this model, the server's "useful" work consists of serving the enquiry frames. The rest of the time, it either is forced to be idle or is carrying protocol frames (i.e., poll or final frames). We include these wasted periods in the server's *walk-time* between queues.

These server walk-times consist of several components. In the remote polling case, for example, the final frame is sent after enquiry frames—if any—from a CC have been transmitted. There is propagation delay, network delay, and delay in the TA_F until the final frame reaches the FEP. The FEP finishes sending any frame it was transmitting and then polls the next CC in the service-order table; i.e., polling has nonpreemptive priority over response frame transmission. The poll is transmitted over the private line from the FEP, and there is propagation delay, network delay, and delay in the TA_C until it reaches the CC. A modem turnaround delay follows, along with some additional delay while the "pipeline" fills up between the CC frame buffer and the transmission line. (This added delay includes CC processing time and modem delay.) Now,

the polled CC begins to transmit the first bit of the first enquiry frame, if there is a frame to send. During all these periods, we say that the server is "walking" between the two queues. If the polled CC has nothing to send, then another walk-time begins immediately.

We denote the walk-time from CC_i to CC_{i+1} (where $1 \leq i < N$) by γ_i , and the walk-time from CC_N to CC_1 by γ_N .

The next section's analysis of the models consists of characterizing, approximately, the distribution of the γ_i (where $1 \leq i \leq N$) for each architecture, and then using some well-known results for the mean delay in a cyclic server queue. Because each enquiry corresponds to one SDLC information frame, inbound enquiry frame delays are equivalent to inbound enquiry delays.

Mean Delay Model. In this section, we describe a model for calculating the mean delay from the time a response from the host enters the FEP to the time the FEP fully transmits the response over the outbound private-line channel.

As described earlier, the FEP *selects* a CC to which it sends responses, using the same service-order table it uses to poll CCs. Thus, strictly speaking, the model on the FEP side should also be a cyclic server model. But this model's walk-time would be zero because the FEP can start sending to the next CC as soon as it finishes sending to the previous one. This implies that, to calculate mean delays, we can model the FEP as a single queue served in first-in, first-out (FIFO) fashion by a single server.⁵

The protocol dictates that the FEP can send, at most, seven frames to a CC before receiving an acknowledgment. In our traffic model, each response fits into three SDLC frames. We can assume that all or none of a response is sent during each select, because the chance is small that more than two responses are queued for a particular CC.

This latter assertion is justified for the traffic model we shall use because the enquiry rate per CC is small. Recall also that, because the CCs are half-duplex, responses cannot be sent to a CC that has been polled. But we assume that, whenever the FEP has responses queued, there is some CC to which it can send a response.

Panel 2. Analysis Notation

N	= number of CCs.
N_{net}	= number of network trunks in the virtual private line.
λ_e	= total arrival rate of enquiries to all CCs (per second).
μ	= tail-circuit line speed (bytes/s).
ϕ	= FEP to TA_F line speed (bytes/s).
v	= trunk speed (bytes/s).
l_{ef}	= enquiry frame length (bytes).
l_{pf}	= poll/final frame length (bytes).
l_{rf}	= response frame length (bytes) = maximum SDLC frame length (bytes).
l_{lapd}	= maximum length of information part of a TA LAPD frame (bytes).
N_{rf}	= number of response frames in a response.
b_{ef}	= l_{ef}/μ .
b_{pf}	= l_{pf}/μ .
b_{rf}	= l_{rf}/μ .
b_{lapd}	= l_{lapd}/μ .
ρ_r	= occupancy of outbound private line on FEP side.
ρ_{net}	= occupancy of each trunk.
t_{lapd}	= l_{lapd}/v .
ϵ	= pipelining build-out delay.
δ_{modswt}	= modem switching (or turnaround) time.
δ_{moddel}	= total transmission delay per modem pair.
δ_{netdel}	= mean network delay for LAPD frame.
$\delta_{netproc}$	= total processing time in switches and adapters.
δ_{prop}	= propagation delay in each direction.
c	= time between a primary device sending a poll and receiving back a final frame, when secondary device had nothing to send and primary device is not sending response frames. (NOTE: This is assumed to be a deterministic quantity, with random components—e.g., network delay—replaced by their means.)
d	= total time for an enquiry frame to be sent over the private access lines in the inbound direction.
g	= delay experienced by a poll waiting for a response frame transmission from TA_C to a CC. This delay occurs if a poll frame follows a response frame, or if a poll has to queue up behind response frames.
h	= delay experienced by a final frame waiting for a response frame transmission from the TA_F to the FEP. This delay occurs if the final frame follows an enquiry frame.
S_e	= mean delay between an enquiry entering a CC and being fully transmitted over the inbound private line on the CC side.
S'_e	= mean delay between an enquiry entering the TA_F and being fully transmitted over the inbound FEP line.
S_r	= mean delay between a response entering the FEP and being fully transmitted over the outbound private line on the FEP side.
S'_r	= mean delay between a response entering a TA_C and being fully transmitted over the outbound line to a CC.
S	= total round trip transport time of an enquiry (i.e., response time minus FEP and host processing times).

These observations and assumptions allow us to model the queuing and transmission of responses with an $M/G/1$ queue, where the arrivals are entire responses and the arrival rate is $\sum_{i=1}^n \lambda_i$. The server models the outbound private line from the FEP. Because response transmissions are interspersed with poll transmissions, the service time of a response is adjusted as described later in the analysis section.

Virtual Circuit Model. The virtual circuit in the network is modeled as a sequence of tandem queues; the server in each queue represents the transmission facility. We assume that the service time is exponential, with a mean equal to the transmission time of a full-size LAPD frame. This assumption is conservative because the average LAPD frame length is less than the maximum frame length, but cannot be determined a priori because the polling traffic is not known. The arrival processes are assumed to be Poisson.

Because the transmission facilities carry traffic other than what is on this virtual private line, we assume that the occupancy of each transmission facility is some suitable value, say 0.8. We also assume that the LAPD frames experience a fixed delay in the protocol adapters and the network switches.

Analysis of the Models

Here, we analyze the models just described to obtain mean delays. Panel 2 contains the notation we will use.

First, we make several approximations in carrying out the analysis:

- We assume that each time a CC is polled, it has, at most, one enquiry to send. This is justified because the total enquiry rate is small and, if the number of CCs is not too small, the enquiry rate per CC is quite small.
- We make the approximation that queuing processes at the FEP and at the CCs are independent. This is justified if the host causes a large random delay.

We also ignore that polling is suspended if the next CC to be polled is being sent response frames. The chance of this happening is small (unless the number of CCs is

small), but the effect is difficult to capture analytically. Simulations have shown that our approximate analysis captures the total round-trip transport delay to within 10 to 15 percent. Further, the analysis is adequate for quantifying the relative performances of the various architectures.

Because of space limitations, we can only briefly outline the analysis for the baseline case. Analyses for the other configurations are similar but have subtle differences that significantly affect performance. An important effect that our analysis tries to capture is the polling delay introduced by the FEP, which must intersperse polls and response frame transmissions. Also, because transmission on the private lines is synchronous, polls occasionally have to wait for completion of a response frame transmission.

Now, suppose the FEP is continuously sending response frames and sends a poll between two response frames. If the CC had nothing to send, then the final frame would arrive back after $c+g$ seconds. But if the CC did have an enquiry frame, then the final frame would arrive back after $c+g+d+h$ seconds. Let

$$m = \lceil \frac{c - b_{pf} + g}{b_{rf}} \rceil$$

and

$$k = \lceil \frac{c - b_{pf} + g + d + h}{b_{rf}} \rceil$$

where $\lceil \cdot \rceil$ is the ceiling function. Also assume that the FEP was continuously sending response frames. In the first case, the FEP would have to poll again after the m^{th} response frame and, in the second case, after the k^{th} response frame.

Let t be a long time interval, and let ρ be the occupancy of the outbound private line. Then, over t , the time for which the outbound line is busy is ρt and, during this time, $\lambda_e \rho t$ enquiries must have been received. Whenever an enquiry frame is received, k response frames follow a poll.

Now, to each of the k response frames, assign $1/k$ of the time to send this poll and argue similarly for the

Panel 3. Parameters for Baseline Case

These parameters are defined in Panel 2. The following values were used in analyzing the baseline case:

$$\begin{aligned} c &= 2b_{pf} + 2\delta_{prop} + \delta_{modest} + \delta_{model} \\ d &= b_{ef} \\ g &= 0 \\ S &= S_e + S_r + 2\delta_{prop} + \delta_{model} \end{aligned}$$

case when no enquiry is sent back from a CC. Then we find that, over time t :

- $k\lambda_e\rho_r t$ response frames have effective transmission time

$$b_{rf} + \frac{b_{pf}}{k}$$

- $\frac{\rho_r t - k\lambda_e\rho_r t \left(b_{rf} + \frac{b_{pf}}{k} \right)}{b_{rf} + \frac{b_{pf}}{m}}$ response frames have effective

$$\text{transmission time } b_{rf} + \frac{b_{pf}}{m}$$

If we divide the left-hand terms by their sum and take a weighted sum of the right-hand terms, we get the average effective response-frame transmission time (which we denote by b_{reff}):

$$b_{reff} = \frac{\left(b_{rf} + \frac{b_{pf}}{m} \right)}{1 + \lambda_e b_{pf} \left(\frac{k}{m} - 1 \right)}$$

Hence, $\rho_r = \lambda_e N_{rf} b_{reff}$.

Similar arguments lead to an approximate distribution for the walk-time in the CC model. Suppose the FEP polls a CC between sending response frames. If the CC sends back an enquiry, then the final frame arrives back while the FEP is sending out the k^{th} response frame. It follows that the walk-time is $kb_{rf} + b_{pf} - b_{ef}$. If the CC sends back no enquiry and the FEP is sending response

frames, then the walk-time is $mb_{rf} + b_{pf}$. But if the FEP is sending only polls and is otherwise idle, the walk-time is simply c . Again if t is a long time period, then the FEP is occupied sending response frames during $\rho_r t$, and $\lambda_e \rho_r t$ enquiries must arrive during this time. Thus we get that, in time t :

- $\lambda_e \rho_r t$ walk-times have length $kb_{rf} + b_{pf} - b_{ef}$.
- $\frac{\rho_r t - \lambda_e \rho_r t (kb_{rf} + b_{pf})}{mb_{rf} + b_{pf}}$ walk-times have length $mb_{rf} + b_{pf}$.
- $\frac{(1 - \rho_r)(1 - \rho_e)^t}{c}$ walk-times have length c .

If we divide each left-hand term by its sum, we get the probabilities that the walk-times have the corresponding values. This is the approximate distribution of each γ_i , where $1 \leq i \leq N$. Of course, the γ_i are dependent but we ignore their correlation for this analysis, which causes our analysis to underestimate the polling delays. At light and heavy loads, in any case, the walk-time will be almost deterministic; hence, this dependence will be weak.

As noted earlier, only seven enquiry frames can be sent during each visit of the server to a CC. Owing to the light load per CC, this limit is not likely to be reached. So we assume that service at each CC is exhaustive. Now, let γ denote the generic random variable for walk-time, with the distribution derived above, and let $\rho_e = \lambda_e b_{ef}$. We find that (see reference 3 or 4):

$$S_e = \frac{(N-1)E\gamma}{2(1-\rho_e)} + \frac{E\gamma^2}{2E\gamma} + \frac{\lambda_e b_{ef}^2}{2(1-\rho_e)} + b_{ef}$$

and the polling rate of the CCs equals $(1 - \rho_e)/E\gamma$.

The above calculation of the effective response-frame transmission time implies that the variability in this time is very small. Ignoring this small variability, we model the outbound transmission with an M/D/1 queue with service time $N_{rf} b_{reff}$ and get:⁵

$$S_r = \frac{\lambda_e \left(N_{rf} b_{reff} \right)^2}{2(1-\rho_r)} + N_{rf} b_{reff}$$

Typically, network trunks are much faster than access lines (e.g., 1.5-Mb/s trunks versus 4800-b/s access lines). So for the end-to-end transaction delay analysis, we shall neglect delay variability in the network. Wherever network delay is needed in the analysis, we simply use the mean delay computed from the M/M/1 model. However, network delay variability is important for calculating pipelining build-out delay. Consequently, in the following, we shall use a typical build-out delay figure that was calculated through a detailed network analysis.

The expressions for c , d , g , and S for the baseline case are given in Panel 3.

This completes the analysis for the baseline case. Analyses for the other configurations involve similar arguments. But each configuration must be carefully analyzed, keeping in mind the details of how polling is done and where queueing occurs.

The general expression for the total round-trip transport delay for all the virtual-circuit-based configurations is:

$$S = S_e + S'_e + S_r + S'_r + 2\delta_{netdel} + 2\delta_{netproc} + 2\delta_{prop} + 2\delta_{model}$$

In the remote-polling configurations, S_e is obtained from the polling model and the walk-times include network delay, propagation delay, and the delays g and h . S_r is obtained from an M/G/1 model for the FEP. S'_e and S'_r include the effects of pipelining or no pipelining.

In the local-polling configurations, S_e is obtained from a polling model for the TA_F. For local polling with simple private-line replacement, S'_r simply indicates whether pipelining is done. But with a high-speed FEP line, because queueing of outbound response frames occurs at the TA_C, S'_r is obtained from an M/G/1 model. Because the TA_C cannot both send to and receive from the only CC on the tail circuit, special care must be taken in analyzing the full fanout case with local polling. So we use a vacation model⁶ to obtain S_e and S'_r in the local polling cases with full fanout.

With a high-speed DDS FEP line, there are no modems between the FEP and TA_F. So δ_{model} is not multi-

Panel 4. Numerical Values for Parameters

μ	= 4800 b/s
ϕ	= 56,000 b/s
l_{ef}	= 80 bytes
l_{rf}	= 270 bytes
N_{rf}	= 3
l_{ef}	= 6 bytes
δ_{modswt}	= 20 ms
δ_{model}	= 24 ms per pair (round trip)
δ_{prop}	= 10 ms (corresponding to 1250 miles)
l_{lapd}	= 40 bytes
LAPD header length	= 6 bytes
N_{net}	= 3 trunks (in virtual circuit)
$\delta_{netproc}$	= 15 ms (2.5 ms per adapter and 5 ms per switch)
ν	= 1.536 Mb/s
ρ_{net}	= 0.8
ϵ	= 30 ms

plied by 2 in the expression for S (see Panel 3), and modem switching delay does not affect the walk-time for the polling model between the FEP and TA_F.

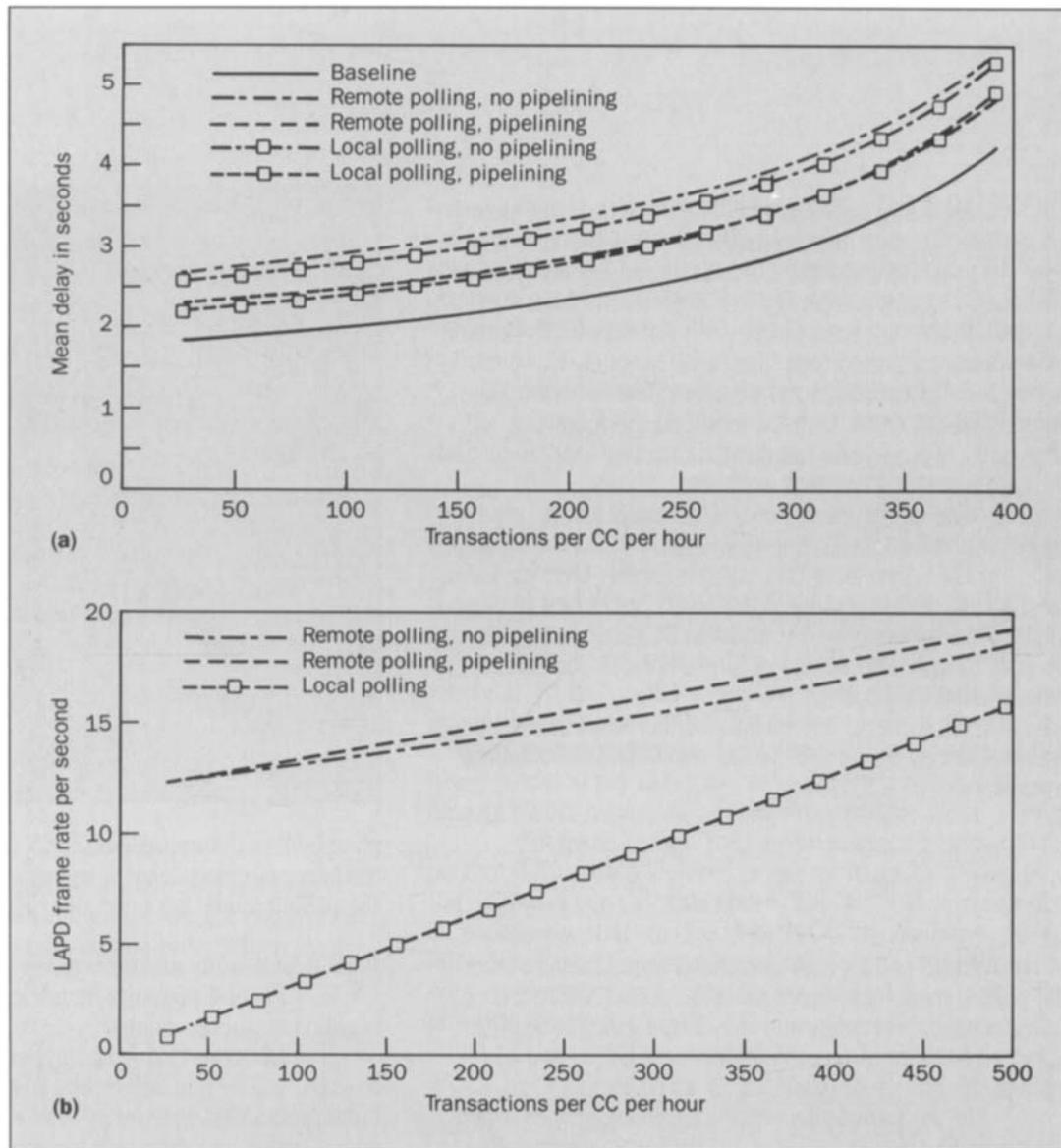
Numerical Results and Discussion

Panel 4 presents numerical results we obtained for certain parameter values.

Observe that l_{ef} and l_{rf} , enquiry and response lengths, are 80 and 270 bytes, respectively. With these enquiry and response lengths on a 4800-b/s line, the outbound traffic limits system transaction capacity to about 520 transactions per CC per hour. In Figures 3 and 4, we plot various performance measures versus the offered load in transactions per CC per hour. Our earlier qualitative discussion should help in understanding these quantitative results.

In Figure 3a, we show the mean round-trip transport delay for the baseline case and simple private-line replacement. We find that all the simple private-line replacement cases yield delays that are uniformly worse than the private-line case. Without pipelining, the delays are about one second more than baseline. Remote polling

Figure 3. Network performance with simple private-line replacement; five cluster controllers (CCs). (a) Mean transport delay; (b) LAPD frame rate in network.



is slightly worse than local polling because, without pipelining, the remote polling rate is also slower.

Pipelining improves mean delay by about 0.5 seconds. Observe that, at high loads, remote polling crosses over and gives slightly better delays than local polling. At low loads, the larger fixed overhead in remote polling makes remote polling delays larger. But at higher loads, double polling—and the inability to take advantage of inbound pipelining—make local polling delays larger.

We can also compare the various private-line

replacement configuration in terms of the amount of load they put on data network resources. In Figure 3b, for example, we show the LAPD frame rate in a virtual circuit that replaces a single 4800-b/s private line. For remote polling, the virtual circuit carries all poll and final frames. This makes the network's LAPD frame traffic significantly higher with remote polling than with local polling.

Owing to smaller polling delays, polling is faster with pipelining. At about 250 transactions per CC per hour, the LAPD frame rate from remote polling is twice

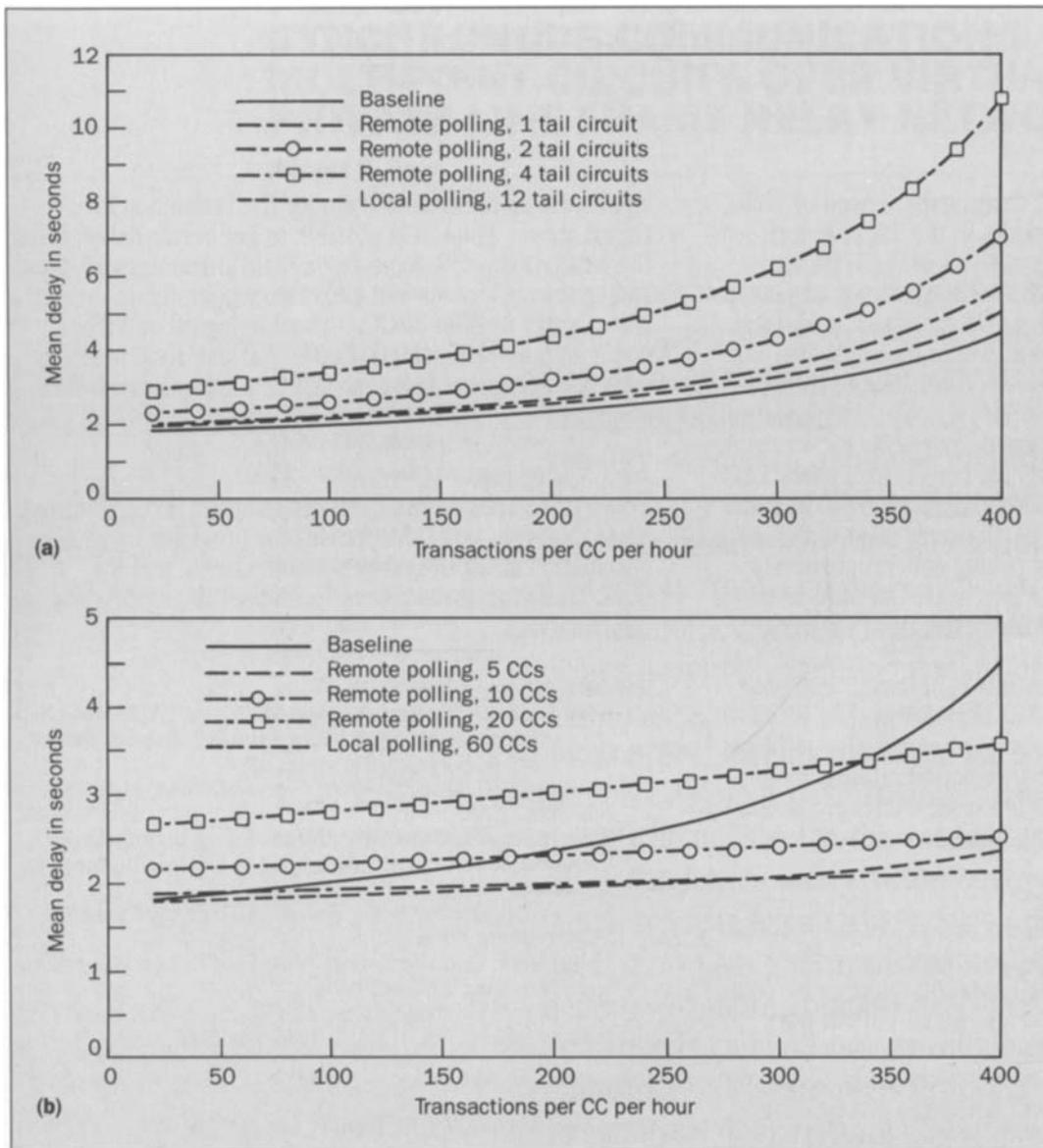


Figure 4. Mean transport delay for logical multipoint at 56 kb/s. (a) Delay with five cluster controllers (CCs) per tail circuit; (b) delay with one CC per tail circuit.

that from local polling. This has obvious implications for network engineering, because the frame rate—not the actual number of bytes transferred—determines the capacity of frame-relay switches.

Although we do not show the results here, the analysis also yields traffic on the virtual circuit in terms of byte rate. Because the poll and final frames are very short, the byte rates offered by remote polling and local polling are not significantly different, except at very light transaction rates.

Multipoint Configurations. We now turn to the analysis of logical multipoint configurations, i.e., those where several tail circuits are multiplexed onto one high-speed line into a FEP port.

In Figure 4a, we show mean transport delays for some cases where the tail circuits are left intact (each tail circuit has five CCs). We assume pipelining is done, even though it does not help significantly. Observe that, even with just one tail circuit, remote polling performs worse than local polling with 12 tail circuits. The main disadvan-

tage of remote polling is the delays experienced by polls that wait behind response frames in the TA_c s. In fact, simulations have shown that our analysis of these remote polling cases is conservative; the delays can be much worse than our analysis shows. On the other hand, local polling is only about 10 percent worse than baseline. But the low-speed tail circuits limit performance to no better than the baseline case.

In Figure 4b, we show delay results for logical multipoint configurations with full fanout, i.e., every CC has its own line into a TA_c port. Even with 60 CCs, local polling now beats the baseline case over most of the range of transaction rates. Remote polling gets progressively worse as the number of CCs increases. But at higher loads, remote polling also yields better delays than the baseline case.

As discussed earlier, this is because, with only one CC per tail circuit, there is a smaller chance that a poll has to wait for response frame transmission to finish. Observe that at high enough transaction rates, remote polling will still beat local polling because, ultimately, double polling will cause local polling delays to increase.

Conclusions

Our results have shown that pipelining reduces delays significantly in the simple private-line replacement configurations. Owing to the limitation imposed by polling all the CCs over at least one low-speed circuit, none of the simple private-line replacement configurations will give better delays than the baseline case. For these configurations, remote polling with pipelining gives delays comparable to local polling with pipelining. Local polling has the advantage of generating less traffic (LAPD frames and bytes) on the network.

Note that these conclusions are based on results for a circuit that uses only terrestrial links, where propagation delays are only a few milliseconds. But if the circuit uses a satellite link, then the propagation delays will be large (hundreds of milliseconds), and local polling may yield significantly better performance.

The real advantage of local polling shows up in the high-speed FEP line (or logical multipoint) configurations. Polling of all CCs over a high-speed line on the FEP side

and fanout on the CC side remove the restriction mentioned above. Thus, it is possible to get better delays than the baseline case. Remote polling still encounters all the low-speed tail circuits and yields very poor delays even with a small number of CCs. Thus, in logical multipoint configurations with a high-speed FEP line, local polling with at least partial fanout should be the implementation of choice.

Acknowledgments

We are grateful to M. Appelbaum, T. C. Kerrigan, H. Q. Nguyen, and J. Medamana for providing us with information about the various architectures, and to B. T. Doshi for useful discussions about the models and their analyses.

References

1. K. A. Boakye, J. C. Kaufeld, and J. W. Palmer, "AT&T Data Networking Architecture," to be published, *AT&T Technical Journal*, Vol. 67, No. 6, November/December 1988.
2. W. Stallings, *Data and Computer Communications*, Macmillan, New York, 1985.
3. S. W. Fuhrmann, "Symmetric Queues Served in Cyclic Order," *Operations Research Letters*, Vol. 4, No. 3, October 1985, pp. 139-144.
4. H. Takagi, *Analysis of Polling Systems*, MIT Press, Cambridge, Massachusetts, 1986.
5. L. Kleinrock, *Queueing Systems*, Vols. I and II, John Wiley & Sons, New York, 1975 and 1976.
6. B. T. Doshi, "Queueing Systems with Vacations—A Survey," *Queueing Systems*, Vol. 1, No. 1, 1986, pp. 29-66.

(Manuscript received May 18, 1988)