# Backbone Network Design and Performance Analysis: A Methodology for Packet Switching Networks

CLYDE L. MONMA AND DIANE D. SHENG

*Abstract*—**This paper describes a packet network design and analysis (PANDA) model which captures the important features of different packet technologies. This model evolved from many iterations with technology developers and network planners over several years. The main contribution is a methodology for designing low-cost backbone packet networks with satisfactory performance which is both practical and useful. This methodology is useful for investigating cost/performance tradeoffs of various network capabilities and components, thus providing a means for identifying potential cost and performance bottlenecks for different packet technologies and to guide capability requirements for new technologies.**

## I. Introduction

THIS paper describes a packet network design and analysis (PANDA) model which captures the important features of different packet switching technologies. This model evolved from many interactions with technology developers and networks planners over several years and is parameterized to facilitate the studying of specific network cost–performance tradeoffs or the assessment of network alternatives. The main contribution of this paper is a methodology for designing packet switching networks and analyzing their performance. This methodology is both practical and useful and has served as the basis of a software package developed by the authors for studying fundamental issues of large packet networks.

Fig. 1 illustrates the generic packet network considered here. Users of the network can be "customers" (individuals or groups) with dial-up or dedicated access capabilities, such as business or residential terminals involved in interactive processing. Also considered are high-traffic users, collectively called "vendors," such as host providers of database services, gateways to other networks, and users with high-speed direct access. The first point of contact with the network for customers is through an access interface. This interface performs traffic concentration, access-protocol-to-internal-network-protocol conversion, and possibly other functions as well. Vendors directly access the switches. A complete discussion of the PANDA model assumptions and definitions of terms is contained in Section II.

Manuscript received October 17, 1985; revised March 20, 1986.

C. L. Monma is with Bell Communications Research, Morristown, NJ 07960.

D. D. Sheng is with AT&T Bell Laboratories, Holmdel, NJ 07733.

IEEE Log Number 8609934.

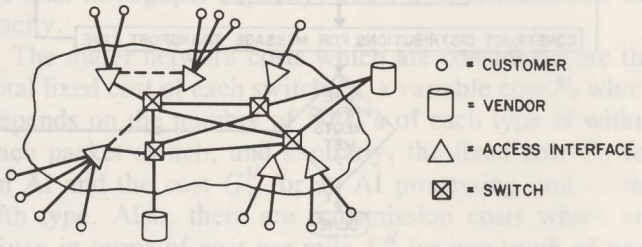



Fig. 1. The network environment.

We address the problem of designing minimum-cost packet networks satisfying certain performance requirements, and we have chosen to use heuristic optimization and approximate analysis methods for general networks in order to find good design solutions for realistic models. Previous work [1]–[6] 1) has made unrealistic assumptions such as exponential traffic flows in order to obtain "exact" solutions quickly or 2) has used detailed, costly simulations of small networks to characterize performance. Indeed, actual simulations often reveal the inadequacies of exponential performance models; see, for example, [7] and [8].

While making design decisions, our algorithm incorporates constraints reflecting the network performance requirements since without such constraints, optimization procedures would load network components to their full capacities in order to minimize costs. These constraints, however, are typically not enough to ensure the configuration of a network that meets all end-to-end performance requirements. The PANDA approach thus iterates between the solution of two subproblems, as shown in Fig. 2. In the optimal packet network design (OPND) subproblem, low-cost network configurations are produced through optimization procedures, and in the packet network performance analysis (PNPA) subproblem, a detailed analysis of the end-to-end performance of those networks is carried out. Network facilities which are performance bottlenecks are identified, and if the network performance requirements are not satisfied, such information is used to adjust the performance constraints incorporated into the optimization procedures. The whole process is repeated until a low-cost network configuration satisfying all the requirements is found.

The OPND methodology configures backbone net-

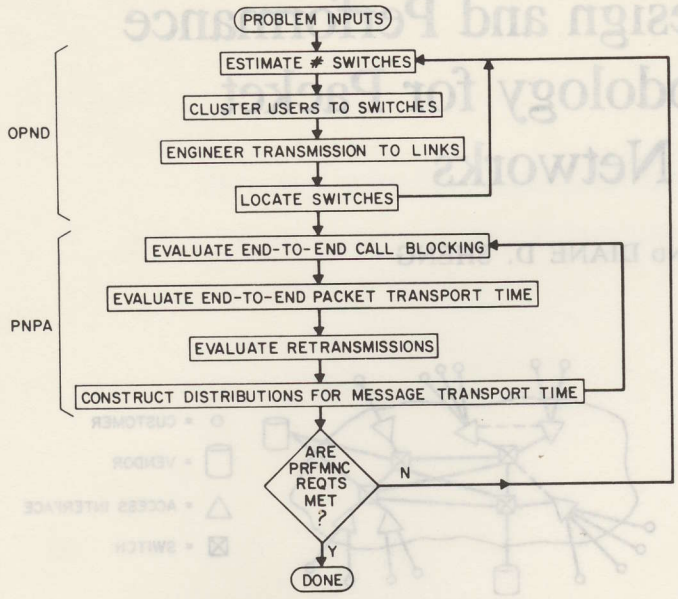



Fig. 2. Principal flowchart of the PANDA approach.

works which minimize total switching and transmission costs. The algorithm uses information describing technology costs and capacities, a projection of user traffic demands, and certain design parameters. The four basic design functions are as follows; see Fig. 2. First, the number of switches required to carry the offered traffic is estimated. Second, clusters of users are formed, one to be assigned to each switch. Third, the number and type of trunks required on each link of the network, and the size of each switch, are determined. Finally, a specific switch location is chosen for each cluster. In contrast to enumerative approaches to finding optimal network designs, this algorithm uses an iterative approach based on repeatedly applying these four steps and converges to a "good" solution quickly.

It should be noted that, while not addressed in this paper, the design of the access portion of the network is also an important task. The major design decisions are where to locate the access interfaces, which customers to assign to which access interfaces, and how to get the customer's data to the access interfaces. Heuristic solution procedures for local access optimization problems have also been developed [9]–[14].

The PNPA algorithm analyzes network end-to-end performance and identifies performance bottlenecks. This algorithm evaluates end-to-end virtual-circuit blocking levels for every pair of network users, and a distribution of end-to-end message transport time for every user-to-user pair and each grade-of-service class. Three kinds of calculations are involved; see Fig. 2. First, an estimate of the probability that a call setup between any pair of users is blocked is constructed. The carried load between every pair of users and related occupancy figures at switches and access interfaces are also calculated. Second, a characterization of the elapsed time encountered by packets in transport from one edge of the network to the other is formed. Finally, the percentage of traffic that must be retransmitted due to errors in transmission over network links or excessive delay in transport is computed. The end-to-end message transport time distributions are then constructed so as to account for retransmitted traffic.

In order for a network design methodology to be most useful, any performance analysis must be convenient to repetitive application. Therefore, in contrast to time-consuming network simulations, the PNPA calculations above are based on analytic formulas. These formulas are derived from general queueing network theory and are complex enough to capture the effects on network performance of "bursty," correlated data traffic flow [15]–[19]. At the same time, however, they are simple enough to be implemented in fast running code.

After one iteration of the PANDA methodology, any PNPA-identified performance bottlenecks are used to update OPND design parameters, and subsequent reapplication of the OPND procedures produces a reconfigured network which attempts to alleviate performance problems. Specifically, the design parameters updated are derating factors which reduce the effective capacity of the network's switches, trunks, and processing units for terminating trunks. This forces the OPND engineering rules to strategically place more equipment, resulting in an improvement in performance and an increase in cost. Thus, PANDA iterations steer the solution towards satisfactory performance while still focusing on cost minimization.

The rest of this paper describes the PANDA methodology in more detail. Section II gives a description of the basic assumptions and definitions adopted. This includes the network management rules, the access interface and switch architectures, the network performance requirements, and the traffic models. Complete descriptions of the OPND methods and the PNPA methods are given in Sections III and IV, respectively. In Section V, we comment on the usage of the PANDA software package. The Appendix lists notation.

## II. Model Assumptions and Definitions

The three key PANDA modeling features are: 1) the problem addressed is the static design of a minimum-cost packet network satisfying certain performance requirements; 2) switching costs dominate transmission costs in the backbone network, and therefore, there is full interconnection among the backbone switches; and 3) in designing the network, no consideration is given to the implementation of specific flow control schemes. Network performance is an equilibrium (steady-state) characterization of performance, and buffers within access interfaces and switches are modeled to have unlimited waiting space.

### A. Network Management

It is assumed that the network adheres to the following routing rules. 1) All virtual circuits between any two customers are set up along the "path(s) of least resistance," i.e., paths involving a minimal number of packet switches. 2) Call setup is spread evenly across all least

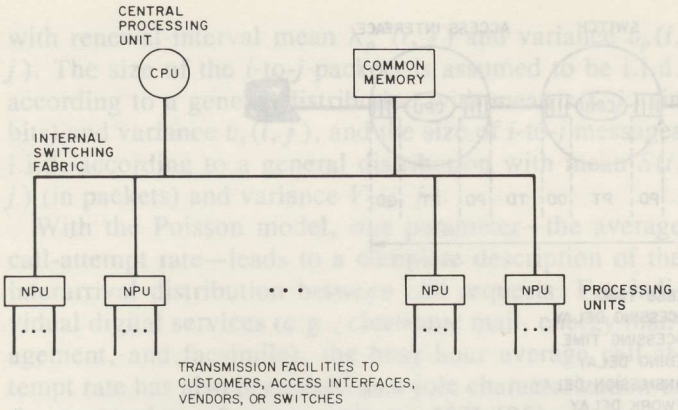



Fig. 3. General access interface/switch architecture.

resistance paths connecting any two customers through load balancing techniques.

### B. Network Components

We adopt a general processor-based architecture for the access interfaces (AI) and the packet switches, as shown in Fig. 3. The fixed part of the switch/AI consists of a central processing unit (CPU), common memory, and an internal switching fabric. Modular processing units terminate trunks and are added as needed. Various trunk types are used, each requiring a corresponding type of processing unit. The capacity limitations are all in terms of an average busy hour.

The CPU performs virtual call setup/teardown and administrative functions, and its real-time capacity potentially limits the overall call attempt rate. For the purposes of this paper, we ignore any such limitation for an access interface, and let $A_S$ denote the number of call attempts per second which is supportable by a switch. The common memory is used to store routing tables and other relevant information for calls in progress, and so imposes a limit on the number of simultaneous virtual circuits that can be handled. Let $C_A$ and $C_S$ denote the virtual-circuit capacity for an AI and switch, respectively.

Messages are formed into (one or more) packets at the first network node. Packets then enter and leave the nodes along a path through nodal processing units (NPU's). In switching packets, each node may perform the following functions: 1) decide to which virtual call an arriving packet belongs, 2) decide to which output transmission facility the packet must be sent, 3) move the packet to the designated outgoing processing unit, 4) hold the packet in a queueing buffer until there is capacity available to transmit it on the designated output facility, and 5) remove the packet from the queueing buffer and transmit it serially on the output facility. The internal switching fabric of an access interface or switch imposes a limit on the overall nodal throughput (in terms of packets per second) and on the number of NPU's one node can support. Let $P_S$ denote the total throughput capacity for a switch and $T_S$ the total number of NPU's supportable by a switch. The NPU's impose a local limit on throughput, call attempt rate, and on the number of allowable terminations.

The PANDA model considers transmission facilities of different types where each type is identified by its transport speed, virtual-circuit capacity, and error characteristics. The NPU's within the switches are distinguished by the type of transmission facilities they terminate. For one trunk of the $H$th type, let $C_R^H$ denote its virtual-circuit capacity, $B_R^H$ the speed (in bits per second), and $v_H$ the bit error rate (BER). It is assumed that the data transport speed $B_R^H$ is simultaneous in both directions. For an NPU terminating type $H$ transmission facilities, let $A_N^H$ denote the number of supportable call attempts per second, $P_N^H$ its total throughput capacity, and $T_N^H$ its terminations capacity.

The major network costs which are considered are the total fixed cost of each switch $F_S$, a variable cost $F_T$ which depends on the number of NPU's of each type $H$ within each packet switch, and similarly, the fixed cost $G_A$ for an AI and the cost $G_T^H$ for an AI processing unit of the $H$th type. Also, there are transmission costs which are given in terms of cost per mile $F_R^H$ for one trunk of type $H$.

Variation of all of the above capacity and cost parameters allows for the modeling of specific packet switching technologies. (See [20] for a discussion of one such packet technology.) Note that setting a capacity parameter to infinity corresponds to suppressing that particular capacity constraint. For example, if there is no limit on the number of simultaneous virtual circuits that a packet switch or a trunk can support (see [21]), then $C_s = \infty$ and $C_R^H = \infty$ in the PANDA model.

### C. Call Setup Blocking and Message Transport Time

When it is not possible to complete a virtual circuit setup due to unavailability of shared network resources, that request for call setup is blocked (lost without generating retrials) by the network. Call setup blocking occurs between two customers when all the virtual circuits are busy on any network component along the least resistance routes connecting the customers.

The transport time of a message is defined to be the elapsed time since the first bit of the message leaves the transmitting customer until the last bit is ready for delivery to the receiving customer by the destination edge of the network (i.e., the final node on the virtual-call path). The main components of message transport time are shown in Fig. 4. The message transport time can be divided into two major components: the access time (AT) it takes for the entire message to reach the sending edge of a network and the network delay (ND) time it takes for the message to go from the sending edge to the receiving edge. Formally, ND is defined as the elapsed time since the last bit of the message arrives at the sending edge of the network until the last bit is ready for delivery by the receiving edge of the network. (Note that the time to traverse the outgoing loop from the network to the customer is not included.)

As each packet of a message travels through the network, it spends time within each of the AI's and switches

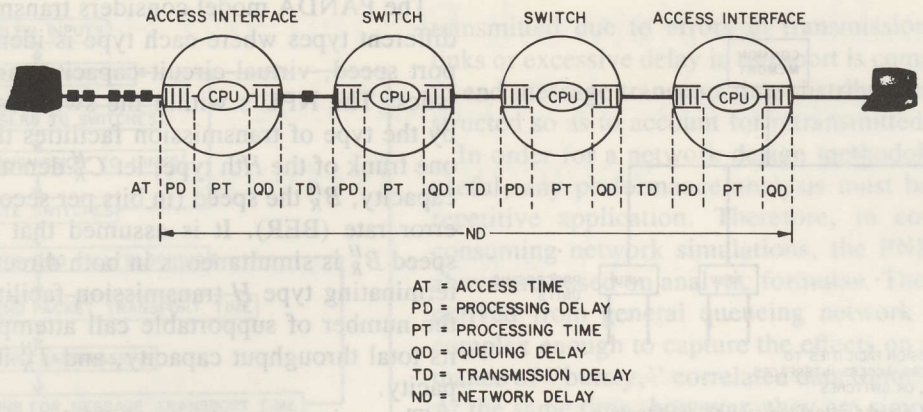



Fig. 4. Basic components in message transport time.

along its virtual-call path. At each node, the basic network delay components for a packet are defined as follows: 1) processing delay (PD)—the elapsed time since the last bit of the packet entered the node at an incoming NPU until the first bit leaves the incoming NPU, 2) processing time (PT)—the elapsed time since the first bit of the packet left the incoming NPU until it enters an outgoing NPU, 3) queueing delay (QD)—the elapsed time since the first bit of the packet entered the outgoing NPU until the first bit is fed onto an output channel, and 4) transmission delay (TD)—the elapsed time since the first bit of the packet is fed onto the outgoing channel until the last bit is fed onto the output channel. Thus, PD and PT together correspond to the nodal time spent on functions 1)–3) in Section II-B, QD is the time spent on function 4), and TD is the time spent on function 5). (We are excluding consideration of propagation delay along network links, i.e., approximately the distance traveled divided by the speed of light.)

In analyzing the above delay components, two modeling assumptions are made about the AI's and switches: 1) each node uses a first-in first-out (FIFO) discipline in moving packets from incoming to outgoing processing units, and 2) each node has one dedicated, infinitely long queueing buffer for each outgoing transmission facility. As a result of the first assumption, the data sequence for packets of the same message is preserved in transport across the network. Therefore, the network delay of a message is identically the network delay of the last packet in the message. As is discussed in Section IV-B, the PANDA methodology uses a characterization of the equilibrium network delay for an arbitrary packet as a characterization of the network delay for the last packet in a message.

The evaluation of message transport times hinges on the analysis of the individual components. Since transmission facilities transport data at fixed speeds, message access time is simply AT = message size/access link speed. Similarly, nodal transmission delay is TD = packet size/outgoing channel speed. We use two parameters to characterize the processing time within a node; one describes the average rate of processing and one describes the variability (potentially zero) associated with the time to process a packet. The parameters associated with an access interface are $\Delta_A$ and $V\Delta_A$, the mean and variance of processing time for an AI. Similarly, for a switch, they are $\Delta_S$ and $V\Delta_S$.

Nodal processing delay and queueing delay are more complex. Both depend on the traffic carried by the node and packet size distributions, as well as the load carried by other nodes and transmission links in the network. In Section IV, a methodology is presented for analyzing nodal processing and queueing delay throughout a packet network.

### D. Retransmissions

In this paper, we limit attention to message retransmission conducted on an edge-to-edge basis.[1] If an error is detected in any packet, at the receiving edge, the entire message is retransmitted over the entire virtual-call path. Also, if a packet takes too much time to traverse its virtual call paths, the entire message is again retransmitted. It is assumed that for a transmission facility of the $H$th type, the probability of any bit being in error is equal to that link's BER $v_H$, and a transmitted packet is considered to be in error if any one bit of the packet is in error.

### E. Network Offered Traffic

The total busy-hour offered traffic is given on a point-to-point basis in each direction between users. Actual traffic loads on transmission links and nodes in the network are derived from these traffic projections and specifications of how traffic is routed through the network.

Consider any two users $i$ and $j$. The arrival of requests for call setups from $i$ to $j$ is modeled as a Poisson process with mean rate $\lambda_c(i, j)$. The holding times of these calls are assumed to be independently and identically distributed (i.i.d.) according to a general distribution with mean $\mu_c^{-1}(i, j)$. The total arrival stream of the packets for transport from $i$ to $j$ on a call is modeled as a renewal process

[1]Of course, not all packet networks operate according to this restriction. It is worth noting, however, that the general technique presented in Section IV-C for analyzing retransmissions over every path through the network can be similarly carried out for each individual link.

with renewal-interval mean $\lambda_p^{-1}(i, j)$ and variance $v_p(i, j)$. The size of the $i$-to-$j$ packets is assumed to be i.i.d. according to a general distribution with mean $s(i, j)$ (in bits) and variance $v_s(i, j)$, and the size of $i$-to-$j$ messages i.i.d. according to a general distribution with mean $S(i, j)$ (in packets) and variance $V_S(i, j)$.

With the Poisson model, one parameter—the average call-attempt rate—leads to a complete description of the interarrival distribution between call requests. For individual digital services (e.g., electronic mail, energy management, and facsimile), the busy-hour average call-attempt rate has also been used as a sole characterization of the generation of setup requests [22]–[26]. A consequence of assuming Poisson call-attempt arrivals is that the level of blocking at individual network components depends on the call holding-time distribution only through its mean holding time. This is the familiar insensitivity property associated with the Erlang loss formula. Projections of busy-hour call expected holding times for individual services are also available [22]–[26].

General (non-Poisson) packet flow processes and general (nongeometric) packet and message size distributions were chosen to better capture the bursty nature of data traffic and the variability (or complete absence of variability) in packet size allowed by some packet switching technologies. Although the complete distributions relevant to data communications traffic are not known, projections of the average message flow, the variance in intermessage intervals, the average message size, and the variance in message size are available for some digital services and data networks [22]–[26]. From these, similar statistics for packet flow and size can be derived for particular packet switching technologies. These projections indicate that Poisson arrival processes and geometric message size distributions are too restrictive for modeling the flow of messages in packet networks. Hence, it is an important feature of the PANDA methodology that its performance analysis incorporates general traffic arrival processes and size distributions.

### *F. Network Performance Requirements*

The two basic measures of network performance considered are the percentage of call requests which are blocked by the network and the distribution of message transport time across the network. The network performance requirements are given separately for different customer service classes. (From these, global network performance objectives are derived, as discussed in Section II-G.) Each customer service class has a characteristic (loop) speed of access to the packet network $L_c$ (in bits per second) for a customer of the $c$th class. Also associated with each class are $v_c$, the BER for its access loop $Z_c$, the average size of its messages (in bits per message), and $T_c$, the time-out threshold for its messages. Any messages originated by a customer of the $c$th service class for which message transport time exceeds $T_c$ are retransmitted. We assume that the proportion of customers for any particular customer service class is the same at every access interface; let $\Theta_c$ denote that proportion of customers belonging to the $c$th customer service class.

The PANDA performance requirements are expressed in the following quantitative terms: 1) a maximum end-to-end blocking of $\beta \times 100$ percent between any two customers, 2) a maximum mean message transport time $\delta_c$ for an originating customer of the $c$th customer service class, and 3) a maximum variance of message transport time $\gamma_c$ for an originating customer of the $c$th customer class. Note that the first requirement is identical for all network customer service classes, and that there is one second and third requirement for each customer service class. The mean transport time objective $\delta_c$ for a customer service class is always less than the time-out threshold $T_c$, and typically is less by at least three standard deviations of the transport time objective. (That is, $\delta_c + 3\sqrt{\gamma_c} \leq T_c$.)

### *G. The PANDA Design Objectives*

In designing networks, the PANDA methodology uses global objectives for call setup blocking from the sending edge to the receiving edge of the network and for the network delay of packets. Subsequently, individual blocking and delay objectives for the network nodes and links are allocated from these edge-to-edge objectives. We choose these global network objectives as follows. Suppose that the network customers include dial-up customers who compete for access ports at the access interfaces, and that the maximum blocking for access across all AI's is $\beta_A$. A global network user-to-user blocking requirement $\beta^*$ is assigned as $\beta^* = \beta - \beta_A$. Similarly, a global network user-to-user mean message transport time requirement $\delta^*$ is constructed as $\delta^* = \min_c (\delta_c - \mathrm{EAT}_c)$ where the minimum is over all customer service classes and $\mathrm{EAT}_c$ is the mean access time for a message of the $c$th customer class. For all messages originated by a class $c$ customer, $\mathrm{EAT}_c$ is simply the message size for class $c$ divided by its access speed, i.e., $\mathrm{EAT}_c = Z_c/L_c$.

Nodal and link blocking objectives are derived by equally allocating the global objective $\beta^*$ among the various network components which are finitely limited in the number of simultaneous virtual circuits they can support along each user-to-user route. That is, if $C_A$, $C_S$, and $C_R^H$ are all finite, then a blocking objective of $\hat{\beta} = \beta^*/7$ is assigned to each node and transmission link since there are at most four nodes (2 AI's and 2 SW's) and three links along any network route. For those network components which are not finitely limited in the number of supportable virtual circuits, $\hat{\beta} = 1$.

For each node, we set one nodal delay objective for the sum of the processing delay and processing time components, and one for the sum of the queueing delay and transmission delay components. In keeping with the design objective of minimizing costs, the allocation of these nodal delay objectives is kept proportional to the relative nodal component costs. The total nodal costs along any user-to-user route is at most $\$\$ = 2 \cdot G_A + 2 \cdot G_T + 2 \cdot F_S + 2 \cdot F_T$. For an access interface, the processing

delay and time objective $\delta_A^P$ is assigned as $\delta_A^P = 2 \cdot G^A/\$\$ \cdot \delta^*$ and the queueing and transmission delay objective as $\delta_A^{QT} = 2 \cdot G_T/\$\$ \cdot \delta^*$. For a switch, we use $\delta_S^P = 2 \cdot F_S/\$\$ \cdot \delta^*$ and $\delta_S^{QT} = 2 \cdot F_T/\$\$ \cdot \delta^*$.

## III. Optimal Packet Network Design (OPND) Methodology

In this section, we describe an approach for designing backbone packet switching networks. Insights into the design problem based on examining a simple case of uniform traffic are obtained in Section III-A. These insights motivate the four major steps of the OPND algorithm which are then described in Sections III-B–III-E. The approach used in Section III-A also provides a means for obtaining a rough estimate of total cast from the total offered traffic.

### *A. Insights from Uniform Traffic Case*

We denote by $t_{ij}$ the traffic *offered* from user $i$ to user $j$, including $i = j$. We denote by $c$ the overall capacity of a packet switch and by $d$ the capacity of a single processing unit. (These traffic values and capacities can be thought of as being in packets/second or call attempts/second.) Other capacity parameters, such as number of terminations, are not considered in this analysis unless explicitly stated otherwise. Let $S$ be the number of switches and $n$ the number of users. The total traffic offered is $T = \Sigma_{i=1}^n \Sigma_{j=1}^n t_{ij}$. For the purposes of the present analysis, we assume that each user is homed onto exactly one switch.

Traffic between users homed onto different switches uses the capacity of both switches. A rough measure of the number of switches $S$ required to handle the total offered traffic $T$ can be obtained by estimating the proportion $p$ of the traffic that uses only one switch. The number of switches is then estimated by the total *effective* traffic divided by the switch capacity or $S = (2(1 - p)T + pT)/c$ since the amount $(1 - p)T$ is double switched while the remainder is single switched. Therefore, $[T/c] \leq S \leq [2T/c]$. ($[x]$ denotes the smallest integer at least as large as $x$.) The lower bound represents no double switching ($p = 1$), whereas the upper bound represents a homing arrangement where all traffic is double switched ($p = 0$).

This argument can be used to estimate the number of switches $S$ required to handle *uniform* traffic (i.e., $t_{ij} = t \equiv T/n^2$ for all points $i$ and $j$, including $i = j$) when the network is *balanced* (i.e., an equal amount of traffic $T/S$ is offered by the points homed onto each of the $S$ switches). In this case, $p = 1/S$ so that $S = (2(1 - 1/S)T + T/S)/c = (2T - T/S)/c$. Solving for $S$ yields $S = [T/c + \sqrt{(T/c)^2 - T/c}]$. The number of switches required in a balanced network with uniform traffic is generally closer to the upper bound than the lower bound on the number of switches needed for general traffic, especially as the load increases.

A balanced network provides good growth potential since it maximizes the spare capacity of the switch with the smallest spare capacity. Unbalanced networks maximize the overall spare capacity among all switches, but they do so by having lots of spare capacity at only a few switches and almost none at others.

For uniform traffic, minimum-cost networks will consist of $(S - 1)$ switches loaded as fully as possible and one (possibly) underutilized switch. (Recall that the major costs are for a switch and its processing units.) This certainly results in the fewest number of switches required and also minimizes the total number of NPU's. The number of NPU's depends on the total effective traffic as seen by all switches in the network, with double-switched traffic counted twice. Thus, relying on cost considerations alone during network optimization can result in unbalanced networks with heavily loaded switches and underutilized ones. In practice, however, such networks may not be very attractive, as they provide little flexibility for growth in traffic demands.

Another implication is that there is no cost savings obtained by placing an additional switch above the minimum number necessary to carry the offered traffic. This follows from the fact that no reduction in NPU's (the other major cost component) can be obtained in this way due to the lack of any communities of interest among the users.

The preceding analysis provides a means of obtaining a rough estimate of the network cost given only the total offered traffic level $T$. By assuming that the traffic is uniform and balanced, the number of switches and the number of NPU's can be estimated from the intraswitch and interswitch traffic. By taking an average access and interswitch link length based on the user locations, the transmission facility costs can also be estimated.

An important insight is the importance of considering balanced versus unbalanced networks from the standpoint of switch and NPU costs. Also, the user and switch locations directly affect the total transmission facility mileage costs. Finally, we note that uniform traffic does not account for communities of interest which may exist among the users. A good design algorithm must also try to make use of these patterns to efficiently build networks.

### *B. Estimating the Number of Switches*

Each user $i$ has a homing option $O_i$ which specifies whether the user is required to be homed onto all switches, exactly one switch, or any subset of switches. Initially, each user $i$ with $O_i =$ "subset" is homed to one switch if its traffic level is relatively low and is homed to all switches if its traffic level is relatively high. We let $TA$ denote the total throughput traffic, i.e., $TA = \Sigma_{i=1}^n \Sigma_{j=1}^n \lambda_p(i, j)$ and we let $TB$ denote the total throughput traffic involving users initially homed to exactly one switch.

The estimate we derive for the number of switches $S$ needed to carry the total throughput traffic is similar to that derived in Section III-A. That is, we assume that the traffic $TB$ is spread uniformly among the users and that the users are balanced over the switches. The effective capacity of a switch depends on the throughput capacity ($P_S$), the derating factor ($f_{PS}$), and the traffic ($TA - TB$) generated by the users homed to all switches and is given

by $f_{PS} \cdot P_s - (TA - TB)/S$ where the number of switches $S$ is to be determined. Now, as in Section III-A, the effective traffic involving users homed to exactly one switch is given by $TB(2 - 1/S)$. The number of switches $S$ is obtained by setting $S$ equal to the effective traffic divided by the effective capacity. Thus, solving a simple quadratic equation and rounding up to an integer, we obtain

$$S = \left\lceil \frac{(TA + TB) + \sqrt{(TA + TB)^2 - 4TB \cdot f_{PS} \cdot P_S}}{2 \cdot f_{PS} \cdot P_S} \right\rceil. \tag{3.1}$$

We note that we have determined only the number of switches at this point. Section III-C determines the homing pattern of users to the $S$ switches and Section III-E assigns locations to the switches after the network is constructed.

### C. Homing Users to Switches

The approach we take is to cluster the users into $S$ clusters according to certain measures of "goodness of fit" based on the ideas of balanced and unbalanced networks, geographical locations, and community of interest as described in Section III-A. Before describing the clustering approach, we introduce five measures for how well a user $i$ fits into a cluster $s$.

The unbalanced network measure $M_U(i, s)$ seeks to favor assigning user $i$ to a cluster $s$ which is most heavily loaded in terms of throughput in order to maintain as unbalanced a network as possible. Specifically, we have $M_U(i, s) = U_p(s)/\hat{M}_U(i)$ where $L_S$ is the set of users currently in cluster $s$, $U_p(s)$ is the throughput utilization of cluster $s$, and $\hat{M}_U(i) = \max_s U_p(s)$. We note that this measure, and all others, is normalized to be between zero and one with a larger value interpreted as being better.

The balanced network measure $M_B(i, s)$ seeks to favor assigning a user $i$ to a cluster $s$ which is least heavily loaded in order to maintain as balanced a network as possible, i.e., $M_B(i, s) = \hat{M}_B(i)/U_p(s)$ where $\hat{M}_B(i) = \min_s U_P(s)$. The balanced and unbalanced measures attempt to capture the issues raised in Section III-A to see what added costs are incurred by maintaining balanced networks. We note that these measures depend only on the cluster $s$ and are independent of the user $i$.

The community of interest measure $M_c(i, s)$ attempts to form clusters of users which collectively send each other more traffic than other subsets of users. This should reduce double-switched traffic and hence reduce cost. We accomplish this by computing the average traffic a user $i$ communicates with other users in a cluster $s$, i.e.,

$$M_c(i, s) = \sum_{j \in L_S} (\lambda_p(i, j) + \lambda_p(j, i))/(|L_s| \cdot \hat{M}_c(i)) \tag{3.2}$$

where $\hat{M}_c(i) = \max_s \Sigma_{j \in L_S} (\lambda_p(i, j) + \lambda_p(j, i))/|L_s|)$.

The geographical measure $M_G(i, s)$ attempts to form clusters of users which are geographically near one another to reduce the cost of transmission facilities. We accomplish this by computing the weighted center of mass $(\bar{x}_s, \bar{y}_s)$ for each cluster $s$ and using the distance to the user $i$ location $(x_i, y_i)$. The center of mass is weighted by the total throughput traffic $\lambda_p(j) = \Sigma_{k=1}^{n} (\lambda_p(i, k) + \lambda_p(k, i))$ generated by a user $j$ in cluster $s$. So we have that $M_G(i, s) = \sqrt{(\bar{x}_s - x_i)^2 + (\bar{y}_s - y_i)^2}/\hat{M}_G(i)$ where $\bar{x}_s = \Sigma_{j \in L_s} \lambda_p(j) x_j / \Sigma_{j \in L_s} \lambda_p(j)$, $\bar{y}_s = \Sigma_{j \in L_s} \lambda_p(j) y_j / \Sigma_{j \in L_s} \lambda_p(j)$, and $\hat{M}_G(i) = \max_s \sqrt{(\bar{x}_s - x_i)^2 + (\bar{y}_s - y_i)^2}$.

The final measure is the random measure $M_R(i, s)$ which assigns a random number drawn uniformly between zero and one independently of the user $i$ and the cluster $s$. This measure is used to break ties and also to introduce new solutions to the process.

Each of these measures has a relative weight or importance $W_U$, $W_B$, $W_C$, $W_G$, and $W_R$, respectively. A cumulative weighted measure of goodness for assigning user $i$ to cluster $s$ is given by $M(i, s) = W_U \cdot M_U(i, s) + W_B \cdot M_B(i, s) + W_C \cdot M_C(i, s) + W_G \cdot M_G(i, s) + W_R \cdot M_R(i, s)$. We now describe the procedure used to home users to switches.

Initially, we have an estimate for the number of switches $S$ with some users already homed to all switches. The remaining users are to be homed to a single switch each. The number of possible configurations is quite large even for a relatively small number of users and switches. The approach taken is to order the remaining users according to importance, e.g., from largest throughput $\lambda_p(i)$ user to smallest, and sequentially assigning each user $i$, in turn, to a feasible cluster $s$ which maximizes $M(i, s)$. By feasible we mean that the total throughput utilization $U_p(s)$ and call attempt utilization $U_c(s)$ for cluster $s$ does not exceed the corresponding effective switch capacities.

Upon assigning the remaining users, we have an initial cluster built up according to the cumulative weighted measure $M$. We note that at the time user $i$ is assigned, the weighted measures only reflect the users assigned before user $i$. In order to overcome any startup bias, we next attempt to adjust the clusters by rehoming users.

The basic step in the rehoming process is to consider each user $i$, currently assigned to the single cluster $s_i$, and to find the cluster $s_i'$ which now maximizes $M(i, s_i')$. Among all users, find the user $i$ which maximizes the rehoming improvement $M(i, s_i') - M(i, s_i)$. This provides a user $i$ which would best benefit from rehoming. If all $M(i, s_i') - M(i, s_i) = 0$, then all users are currently assigned to the "best" cluster and we are done. If, on the other hand, there is a position improvement, we want to consider rehoming user $i$ from cluster $s_i$ to cluster $s_i'$. If user $i$ can be feasibly assigned to cluster $s_i'$, then we do so. If not, we find a user $k$, currently in cluster $s_i'$, which can be feasibly exchanged with user $i$ and which gives the maximum (positive) exchange improvement $M(i, s_i') - M(i, s_i) + M(k, s_i) - M(k, s_i')$. If no such user $k$ can be found, then another user maximizing the rehoming improvement is considered for rehoming. This basic step is repeated a fixed number of times to make up one iteration. If, after an iteration, the resultant network has a smaller

cost than the network at the previous iteration, then we repeat an iteration. This continues until there is no cost improvement after an iteration.

Recall that certain users $i$ have a homing option $O_i$ which allows belonging to any subset of clusters. The final step of the homing process adds a user (which initially appeared in only one cluster) to other potentially beneficial clusters or removes a user (which initially appeared in all cluster) from relatively unimportant clusters. Again, the measure of importance used to select the potential clusters to add or drop is the cumulative weighted measure for user $i$ to belong to cluster $s$. A user is actually added or dropped only if it results in a cost improvement.

Upon the completion of the steps outlined in this section, we have a desired clustering that seeks to minimize total switching and transmission costs through a weighted measure. This homing pattern together with the routing rules described in Section II-A determine the throughput and call-attempt utilization levels on a link from user $i$ to switch $s$ ($U_p(i, s)$, $U_A(i, s)$), on a link between switches $s_1$ and $s_2$ ($U_p(s_1, s_2)$, $U_A(s_1, s_2)$), and at a switch $s$ ($U_p(s)$, $U_A(s)$). These values are used in Section III-D to engineer the transmission facilities on each link and the NPU's at each switch.

### D. Engineering Trunks and Switches

In section III-D1), we describe the rules used to engineer the number of trunks on each access link from user $i$ to switch $s$, ($R_{is}$), and each interswitch link between switches $s_1$ and $s_2$, ($R_{s_1s_2}$). One type of trunk is chosen for all access interface users to switches (type $H_A$), one type for all vendor users to switches (type $H_V$), and one type for all interswitch links (type $H_s$). Section III-D2) describes how the number of NPU's is determined at each switch.

The number of access NPU's ($N_S^{H_A}$) at switch $s$ can be obtained by looking at the access throughput traffic ($U_p(i, s)$), call-attempt traffic ($U_A(i, s)$), and number of access trunks ($R_{is}$), dividing by the effective capacities $f_{PN}^{H_A} \cdot P_N^{H_A}$, $f_{AN}^{H_A} \cdot A_N^{H_A}$, and $f_{RN}^{H_A} \cdot R_N^{H_A}$, respectively, and taking the maximum rounded up to the nearest integer. That is, $N_S^{H_A} = \max [\Sigma_{\text{AI}i} U_p(i, s)/(f_{PN}^{H_A} \cdot P_N^{H_A}), \Sigma_{\text{AI}i} U_p(i, s)/(f_{AN}^{H_A} \cdot A_N^{H_A}), \Sigma_{\text{AI}i} R_{is}/(f_{RN}^{H_A} \cdot R_N^{H_A})]$. A similar expression can be derived for vendor NPU's by replacing $H_V$ for $H_A$ and summing over all vendors $i$ rather than AI's $i$.

*1) Engineering Transmission Facilities:* For a given transmission facility type, engineering any particular link is handled by determining the minimal number of trunks needed such that acceptable performance on that link is provided. Two measures of link performance are considered: the blocking of virtual circuit setup across that link and the expected delay encountered in the transport of packets across the link in each direction. Acceptable performance is specified by the blocking objective of $\hat{\beta}$ for all links and the queueing-and-transmission delay objectives of $\delta_A^{QT}$ and $\delta_S^{QT}$ for access and interswitch links, respectively; see Section II-G.

Consider the access link from user $i$ to switch $s$. The number of trunks of type $H$ needed so as to satisfy the link blocking objective $R_{is}^{bH}$ is calculated through use of the standard Erlang blocking model $M/G/N/N$ [16]. The call arrival rate in the model is $\lambda_c = \Sigma [\lambda_c(i, j) + \lambda_c(j, i)] \cdot n(i, j, s)/n(i, j)$ where the summation is over all users $j$ such that $i$ and $j$ communicate via switch $s$ and where $n(i, j)$ and $n(i, j, s)$ denote, respectively, the number of (least resistant) paths between $i$ and $j$ and the number of paths between $i$ and $j$ via $s$. The mean holding time is $\mu_c^{-1}$ where $\mu_c^{-1} = (1/\lambda_c) \Sigma [\lambda_c(i, j)/\mu_c(i, j) + \lambda_c(j, i)/\mu_c(j, i)] \cdot n(i, j, s)/n(i, j)$ and the summation is as before. The Erlang loss formula $B(N, \alpha) = (\alpha^N/N!)/\Sigma_{j=0}^{N} (\alpha^j/j!)$ with $N = C_R^H$ and $\alpha = \lambda_c/\mu_c$ is then the call blocking probability for the link. We determine $R_{is}^{bH}$ by first finding $N^* = \min \{N > 0: B(N, \alpha) \leq \hat{\beta}\}$ and then simply dividing by the virtual circuit capacity of a trunk and rounding up, $R_{is}^{bH} = \lceil N^*/C_R^H \rceil$.

We follow an iterative procedure to calculate $N^*$, but instead of cumbersome numerical calculation of $B(N, \alpha)$ we use asymptotic and approximate representations [17] which result in rapid approximation of $B(N, \alpha)$ even for very large $N$. These representations, however, do not yield approximations of arbitrarily high accuracy. Instead, their accuracy depends on the specific values of $N$ and $\alpha$ where it is their relative size $c = \alpha - N/\sqrt{N}$ which is important. For the large values of $N$ and $\alpha$ relevant to the PANDA model, these approximations to $B(N, \alpha)$ are within five percent accuracy [17].

To calculate $N^*$: 1) set $N_{\text{UPP}} = \alpha + c_{\text{UPP}} \sqrt{\alpha}$ as an upper bound for $N^*$, 2) set $N_{\text{LOW}} = \alpha + c_{\text{LOW}} \sqrt{\alpha}$ as a lower bound for $N^*$, and 3) iteratively search for $N^* \in [N_{\text{LOW}}, N_{\text{UPP}}]$ such that

$$B(N^*, \alpha) \sim \left[ a_0(c) \sqrt{N^*} + a_1(c) + \frac{a_2(c)}{\sqrt{N^*}} \right]^{-1} = \hat{\beta} \tag{3.3}$$

where $c$ is defined as before. The coefficients $a_0(c)$, $a_1(c)$, and $a_2(c)$ are found through interpolation of Table I in Jagerman [17]. Solving for $N$ as a function of $c$ leads to taking $N_{\text{UPP}}$ and $N_{\text{LOW}}$ in the forms indicated above as bounds on $N^*$ where the coefficients $c_{\text{UPP}}$ and $c_{\text{LOW}}$ are dependent on the desired link blocking level $\hat{\beta}$ and offered load $\alpha$. For large data networks where the relevant ranges of $\alpha$ and $\beta^*$ may be as large as [200, 15 000] and [0.001, 0.05], respectively, $c_{\text{UPP}} = 2$ and $c_{\text{LOW}} = 0$ should be used in the above procedure.

The number of trunks of type $H$ needed to satisfy the access link delay objective $R_{is}^{dH}$ is calculated through use of a queueing delay model in which the "customers" are the packets arriving for transport across the link and "service time" is the transmission time. In particular, a system of $N$ parallel general single-server queues $GI/G/1/\infty$ is used to model a link comprised of $N$ trunks. The arrival process in the user-$i$-to-switch-$s$ model is a renewal process with rate $\lambda = \Sigma_j \lambda_p(i, j) \cdot n(i, j, s)/n(i, j)$ and renewal-interval variance $\gamma$. It is assumed that the average

arrival rate to each individual queue is $\lambda/N$. The service times at each of the individual queues are i.i.d. with mean length $\mu^{-1} = (1/B_R^H) \sum_j (\lambda_p(i,j)/\lambda) \cdot s(i,j) \cdot n(i,j,s)/n(i,j)$ and variance

$$\eta = \left[ (1/(B_R^H)^2) \sum_j (\lambda_p(i,j)/\lambda) \cdot (n(i,j,s)/n(i,j)) \cdot [v_s(i,j) + s^2(i,j)] \right] - (1/\mu)^2.$$

Thus, $R_{is}^{dH}$ is the minimal number $N$ such that the mean waiting time spent in each of the $N$ parallel $GI/G/1/\infty$ queues representing $i$-to-$s$ transport and each of the $N$ parallel queues representing $s$-to-$i$ transport is less than the objective $\delta_A^{QT}$.

The above system of $N$ parallel, single-server queues was chosen over a system of one $N$-server queue as the model to analyze each direction of the transmission link so as to better capture the nature of virtual-call packet switching. Once a call has been set up across a transmission link, a specific trunk on that link has been designated for transporting all the packets involved in that call.

Exact solutions for the steady-state behavior of $GI/G/1/\infty$ queues do not exist; however, approximations are available. The approximation used here [18], [19] for the mean waiting time (before service begins) for each $GI/G/1/\infty$ queue above is

$$EW \sim \frac{\rho}{(1-\rho)\mu} \left[ \frac{c_a^2 + c_s^2}{2} \right] g(\rho, c_a^2, c_s^2) \qquad (3.4)$$

where $\rho = \lambda/N\mu$, $c_a^2 = \lambda^2\gamma$ is the squared coefficient of variation for interarrivals, $c_s^2 = \mu^2\eta$ is the squared coefficient of variation for service, and

$$g(x,y,z) = \begin{cases} \exp\left\{ -\dfrac{2(1-x)}{3x} \dfrac{(1-y)^2}{y+z} \right\} & \text{if } y \le 1 \\ 1 & \text{if } y \ge 1. \end{cases}$$

For the sake of computational expediency, the PANDA methods use the same "average" variance parameter $\gamma$ for both directions across all access links. That global average for all user-to-user pairs is determined by

$$\gamma = \left[ \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\lambda_p(i,j)}{TA} \left[ v_p(i,j) + \left( \frac{1}{\lambda_p(i,j)} \right)^2 \right] - \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\lambda_p^2(i,j)}{TA} \right)^{-2} \right] \qquad (3.5)$$

where $TA$ is, as before, the network's total throughput traffic.

For the user-$i$-to-switch-$s$ direction, let $N_{is}^* = \min \{N > 0: EW \le \delta_A^{QT}\}$. The solution for $N_{is}^*$ is found by iterating on values for $N$ and repeatedly evaluating $EW$. The procedure is: 1) set $N_{\text{UPP}} = \lambda/\mu + (\lambda^3\gamma + \lambda\mu^2\eta)/2\mu^2\delta_A^{QT}$ as an upper bound for $N$, 2) set $N_{\text{LOW}} = \lambda/\mu$ as a lower bound for $N$, and 3) iteratively search for $N \in [N_{\text{LOW}}, N_{\text{UPP}}]$ such that $EW = \delta_A^{QT}$ where $EW$ is calculated by the approximation (3.4). Similarly, $N_{si}^*$ is found for the switch-$s$-to-user-$i$ direction. The access link delay objective is then satisfied at $R_{is}^{dH} = \max \{N_{is}^*, N_{si}^*\}$ and the total number of trunks of type $H$ needed is $R_{is}^H = \max \{R_{is}^{bH}, R_{is}^{dH}\}$.

Application of the two procedures above also produces $R_{s_1,s_2}^H$, the minimal number of type $H$ trunks needed between switch $s_1$ and switch $s_2$. Determining the final number of trunks on each access link $R_{is}$ and each interswitch link $R_{s_1s_2}$ is handled by optimizing over the different types of transmission facilities. All access links and interswitch links are engineered for each type $H$; and final selection of the type to link access interfaces to switches $H_A$, the type to link vendors to switches $H_V$, and the type to link switches $H_S$ is made so as to minimize total link costs.

*2) Switch Engineering:* The NPU's at a particular switch $s$ are treated as a pool of resources. That is, the process of assigning individual trunks to specific NPU's is not considered. Rather, the total usage on all trunks is compared to the NPU capacities to determine the number of NPU's needed. Recall that NPU's are of type $H_A$, $H_V$, and $H_S$ for access interface trunks, vendor trunks, and interswitch trunks, respectively. We wish to compute $N_S^{H_A}$, $N_S^{H_V}$, and $N_S^{H_S}$, the number of NPU's of each type at each switch. For interswitch NPU's,

$$N_S^{H_S} = \max \left[ \sum_{s_1} U_p(s,s_1) \Big/ (f_{PN}^{H_S} \cdot P_N^{H_S}), \sum_{s_1} U_A(s,s_1) \Big/ (f_{AN}^{H_S} \cdot A_N^{H_S}), \sum_{s_1} R_{ss_1} \Big/ (f_{RN}^{H_S} \cdot R_N^{H_S}) \right]$$

and similar computations determine $N_S^{H_A}$ and $N_S^{H_V}$.

Finally, we note that if the total number of NPU's at any switch $s$ exceeds the switch capacity, i.e., $N_S < N_S^{H_A} + N_S^{H_V} + N_S^{H_S}$, then the solution is infeasible and the OPND algorithm starts anew to reconfigure another network with $(S+1)$ switches.

### E. Locating Switches

Having determined the *number* $S$ of switches (Section III-B), the clustering of users into $S$ groups (Section III-C), and the number and type of trunks and NPU's (Section III-D), the final OPND step is to choose a switch location for the users in each cluster to be homed to. The cost consideration here is to minimize total mileage costs. Generally, the access trunk costs will tend to dominate the interswitch trunk costs. As a result, we choose to locate a switch at a feasible site closest to the center of mass $(\bar{x}_s, \bar{y}_s)$ of the users in a cluster $s$. Specifically, we weigh the $(x, y)$ location of each user by the number of access trunks $R_{is}$ between the user $i$ and cluster $s$. Thus, $\bar{x}_s = \Sigma_{i \in L_s} R_{is} x_i / \Sigma_{i \in L_s} R_{is}$ and $\bar{y}_s = \Sigma_{i \in L_s} R_{is} j_i / \Sigma_{i \in L_s} R_{is}$. This tends to move the switch location closer to larger users, thus reducing transmission costs.

## IV. Packet Network Performance Analysis (PNPA) Methodology

In this section, we describe in detail a methodology for evaluating the performance of packet networks. The three major steps of the analysis are: 1) to estimate call block-

ing for any particular pair of customers, 2) to characterize the network transport time for packets, and 3) to measure the impact of retransmitted traffic and incorporate the results in the construction of message transport time distributions for every pair of network customers. Sections IV-A–IV-C describe these PNPA steps, respectively.

### A. End-to-End Call Blocking

Consider any two network users $i$ and $j$, and let $b(i, j)$ denote their end-to-end blocking level. Once having calculated blocking levels for all pairs of users, they can be checked against the network blocking requirement $\beta$, i.e., $b(i, j) \leq \beta$.

As a result of the network routing rules used in the PANDA model, $b(i, j)$ is the average of end-to-end blocking over all paths of least resistance connecting $i$ and $j$. That is, $b(i, j) = \Sigma b_{ij}^p / n(i, j)$ where $b_{ij}^p$ is the blocking between users $i$ and $j$ over specific path $p$ and where the summation is over all $n(i, j)$ $i$-to-$j$ paths $p$. The blocking $b_{ij}^p$ is evaluated in terms of the blocking levels at the individual components along the path. The component blocking levels are calculated for each AI, switch, and transmission link which is finitely limited in its number of supportable virtual circuits through application of the $M/G/N/N$ queueing model [16] described in Section III-D1). For each AI, $N = C_A$ and for each switch, $N = C_S$. For each transmission link made up of $m$ trunks of type $H$, $N = m \cdot C_R^H$. A component's call setup arrival rate and the mean call duration times are determined by averaging, as weighted by traffic usage, the user-to-user arrival rates and holding times across all user pairs with a path utilizing that component.

Let $b_1, b_2, \cdots, b_z$ denote the blocking levels for the individual network component, where the AI's, switches, and transmission links have been arbitrarily numbered $1, 2, \cdots, z$. (For those components with infinite virtual circuit capacity, $b_k = 0$.) End-to-end blocking between users $i$ and $j$ over a specific path $p$ is then standardly evaluated as $b_{ij}^p = 1 - \Pi_{r=1}^n (1 - b_{N_r}) \Pi_{s=1}^l (1 - b_{L_s})$, in which $N_1, \cdots, N_n$ are the component numbers of the processing nodes and $L_1, \cdots, L_l$ are the component numbers of the transmission links on that path through the network. Note that in using the above procedure to calculate component and end-to-end blocking levels, stochastic independence between network components is assumed. That is, each component has been assigned a blocking level that is independent of the blocking levels at all other components in the network. Additionally, each individual component has been assigned an offered load which is a direct mix of the original user-to-user offered traffic levels, and not a mix of the actual carried loads at neighboring network components. That this procedure for evaluating end-to-end blocking is conservative, i.e., overestimates $b(i, j)$, has recently been shown by Whitt [27].

### B. Network Packet Transport Time

As discussed in Section II-C and illustrated in Fig. 4, message transport time is the sum of the access time (AT) for the message and the network delay of the last packet in the message. Furthermore, the network delay of the message's last packet is the sum of the various processing delays (PD), processing times (PT), queueing delays (QD), and transmission delays (TD) seen by this last packet as it is transported across the network. The PNPA procedure for calculating the distribution $F_{ijc}(t)$ of message transport time from user $i$ to user $j$ for an originating customer of service class $c$ is basically as follows: 1) Evaluate all PD, PT, QD, and TD components in the packet network for their steady-state characterizations, 2) for each least resistant path $p$ connecting $i$ and $j$, approximate $F_{ij}^p(t)$, the network delay of the last packet in an $i$-to-$j$ message over $p$, by the sum of PD, PT, QD, and TD components along $p$, 3) calculate $F_{ij}(t)$, the network delay distribution from $i$ to $j$, 4) note that for all messages originated by service class $c$, AT is simply class $c$'s message size divided by its speed; $AT_c = Z_c/L_c$, 5) finally, $F_{ijc}(t - AT_c)$.

Once having constructed $F_{ijc}(t)$, its mean is checked against $\delta_c$, the $c$th class' mean message transport time requirements. The variance of the $F_{ijc}(t)$ distribution is checked against the variance requirement $\delta_c$.

Implicit in step 2) above is the assumption that the network delay for the *last* packet in a message is approximated by the distribution of the network delay of an *arbitrary* packet. If messages contain a geometrically distributed number of packets, then these two distributions indeed coincide under FIFO switching [28]. If the number of packets in a message follows some other distribution, then a stochastic ordering of the delay distribution for the last packet in a message and the delay distribution for an arbitrary packet may exist [29]. Therefore, alternative approximations to the delay distributions for the last packet in a message might be constructed by appropriately inflating or deflating the delay distributions for an arbitrary packet, depending on the message-size distributions. Construction of the network delay distributions for specific paths is discussed in Section IV- B5).

The most critical and computationally intensive task in the above is step 1). This is accomplished through application of an open network-of-general-queues model, as discussed in Section IV-B1) below. Approximate solutions for the steady-state behavior of such general queueing networks provide the evaluation of the packet network delay components. It is not possible, however, to obtain solutions for very large queueing networks; therefore, further network decomposition is needed. An efficient decomposition method corresponds to decomposing the packet network into its different levels of concentration and subsequently evaluating the individual PT, PD, OD, and TD components for each level of the network separately, as discussed in Sections IV-B2)–IV-B 4). First, the delay components for the network's access portion corresponding to the transport of traffic directly between AI's and from the users to the switches are analyzed. Second, the delay components for the backbone switching portion are analyzed, using the results of the

access portion analysis to characterize the arrival of traffic to the backbone network. Third, the delay components for the access portion corresponding to the transport of traffic from the switches to the AI's and vendors are analyzed, again incorporating the results of the previous analyses.

*1) The PNPA Open Network-of-Queues Model:* To capture the network effects of packet transport time along any path through the network, the entire packet network configuration is modeled as an open network of *GI/G/* $1/\infty$ queues. In this network of queues, there is one queueing station for each AI and switching node in the packet network and one for each output trunk at every AI and switching node. An explicit illustration of the network of queues corresponding to the small network pictured in Fig. 5(a) is given in Fig. 5(b), which we now explain.

These queueing stations which represent interfaces and switches are called "processing" *P* stations, while those which represent output channels are called "transmission" *T* stations. Nodal transmission delay in the packet network is then the service time associated with the *T* stations in the network of queues. Nodal queueing delay is the time spent waiting for service by a customer at the *T* stations in the network of queues. Nodal processing time is the service time associated with the *P* stations and nodal processing delay is the time spent waiting for service at the *P* stations.

In Fig. 5(b), the labeled circles symbolize the different queueing stations in the network of queues. We denote by *X* the queueing station corresponding to the processing node. We denote by *XY* the queueing stations corresponding to the holding buffer at node *X* for the trunk from *X* to *Y*. The subscript *m* in $XY_m$ denotes the holding buffer for the *m*th trunk from *X* to *Y*. The directed arcs connecting the queueing stations represent the flow of packets between nodal processors and holding buffers. Note that there are four queueing stations associated with the two trunks between switches *A* and *C*.

By evaluating the probability distribution of the total time a customer spends in the PNPA network of queues (sojourn time), distributions for network packet transport time are estimated. To date, complete analytical solutions for the steady-state behavior of such general queueing networks do not exist. However, good approximations are available. The PNPA algorithm uses the QNA approximation [18], [30] which, because of its noniterative algorithmic approach, is computationally fast running. Other approaches have been to iteratively arrive at an approximation [31], or to replace the approximate analysis of a general queueing network with the exact analysis of a Markovian queueing network [32], or to analyze the general queueing network under some limiting conditions [33].

The general procedure of QNA is to represent all the arrival processes and service-time distributions within the network of queues by two parameters, one to describe the rate and one to describe the variability. The congestion at each queueing station is then described as a function of these parameters. Arbitrarily numbering the queueing stations in the networks of queues representation of the packet network, the two parameters characterizing the service times at queueing station $k$ are the mean service time $\tau_k$ and the squared coefficient of variation $c_{sk}^2$. For each arrival process, the two parameters used are associated with fitting a renewal process model to the arrival process. For the external arrival process to queueing station $k$ (customers entering the network at station $k$), the parameters are the external arrival rate $\lambda_{0k}$ and the squared coefficient of variation of the external renewal interval $c_{0k}^2$. For the overall arrival process to queueing station $k$ (including customers arriving at $k$ from other queueing stations), the parameters are the overall arrival rate of customers $\lambda_k$ and the squared coefficient of variation of the interarrival times $c_{ak}^2$.

The parameters $\tau_k$, $c_{sk}^2$, $\lambda_{0k}$, $c_{0k}^2$, $\lambda_k$, and $c_{ak}^2$ are collectively solved for in the QNA approximation from input information about different types of customers in the network of queues, as discussed in Whitt [18]. A customer type has a specific route (sequence of queueing stations visited) through the network of queues. Each type arrives at the first queueing station on its route, according to a process which is characterized by $\hat{\lambda}_q$, the arrival rate for type $q$ and $\hat{c}_q^2$, the squared coefficient of variation of the external renewal interval for type $q$. Each type $q$ has a specific service-time distribution at each queueing station on its route, which is characterized by $\hat{\tau}_{ql}$, the mean service time, and $\hat{c}_{sql}^2$, the service-time squared coefficient of variation for type $q$ at the $l$th queueing station on its route.

In the PNPA network of queues representation, a customer type corresponds to a specific user-to-user traffic flow over a particular virtual-call path through the packet network. For example, there are two customer types in the network of queues in Fig. 5(b) representing user-1-to-user-3 traffic in the sample network in Fig. 5(a). The two types represent the choice among two trunks over the transmission link between switches *A* and *C*. Similarly, there are three customer types representing the user-1-to-user-5 traffic, accounting for the three distinct min-hop virtual-call paths connecting users 1 and 5, and 36 customer types in all.

Note that the higher the number of nodes and links and user-to-user paths in the packet network, the much higher the number of queueing stations and customer types in the resulting network-of-queues model. (The number of queueing stations roughly grows cubically with the number of nodes and links.) For large packet networks of the size considered by the PANDA model (on the order of 100 users, 15 vendors, and 15 second-level packet switches), it is therefore efficient and necessary to decompose the analysis of the network of queues. Each section of the decomposed network of queues is then analyzed through separate, yet sequentially coordinated, applications of the QNA approximation.

*2) High-Usage Link and Access Interface to Switch Delays:* The delay components for the network's access portion are evaluated by analyzing the section of the de-

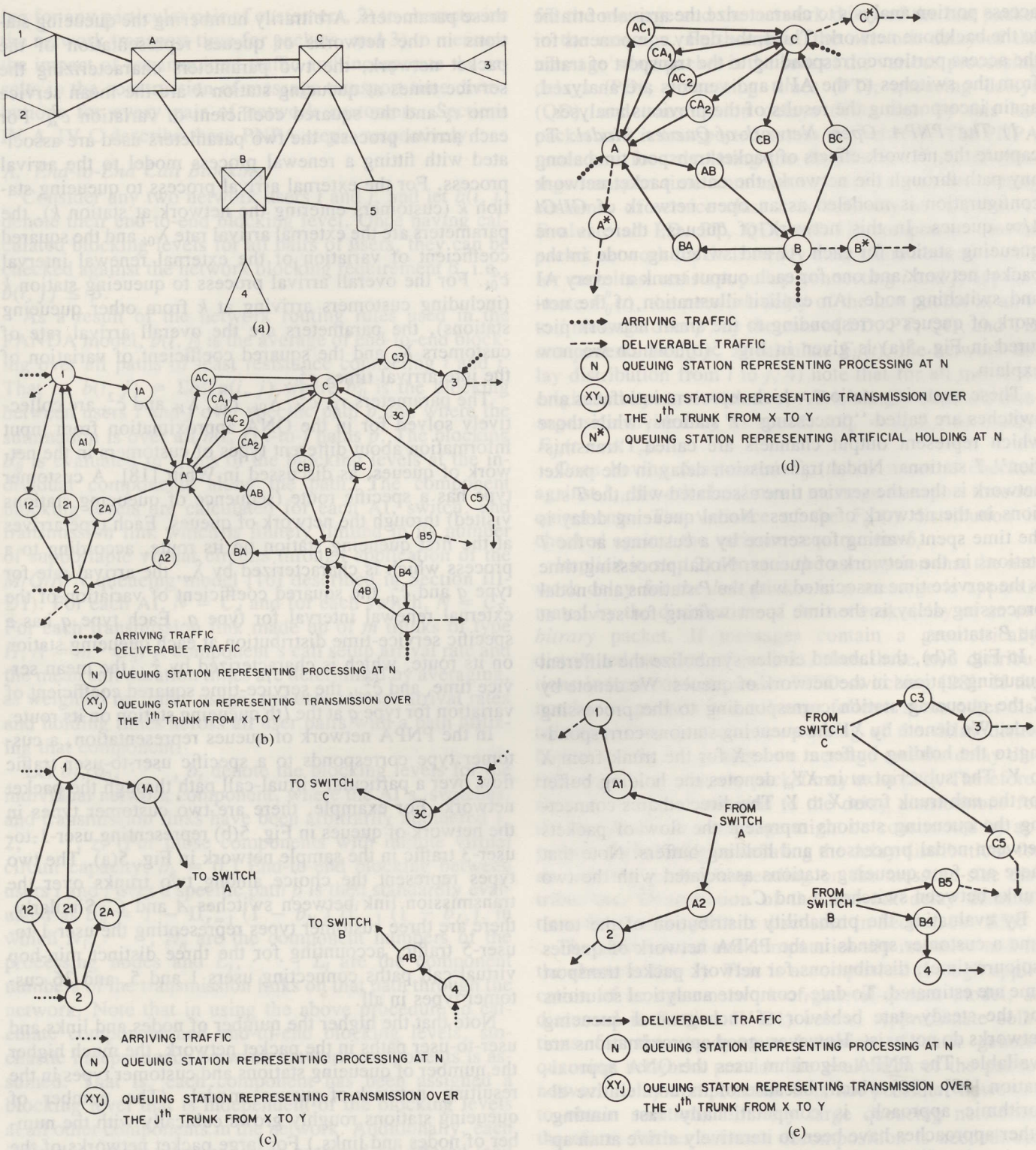



Fig. 5. (a) Small network example. (b) Corresponding network of queues model. (c) Corresponding subnetwork of queues for high-usage link and AI to switch delays. (d) Corresponding subnetwork of queues for inter-switch delays. (e) Corresponding subnetwork of queues for switch to user delays.

composed network of queues containing the queueing stations representing processing at the AI's, transmission between AI's, and transmission from AI's to switches. Fig. 5(c) shows this subnetwork of queues from the entire network given in Fig. 5(b).

Let the QNA customer types in the resulting subnetwork of queues be arbitrarily numbered. The last queueing station on each type's route is a $T$ station, which represents either transmission over a high-usage link or transmission over an access link from an AI to a switch. Con-

sider type $q$, corresponding to the flow of traffic from AI $i$ to AI $j$ over the $n$th trunk of the high-usage link between $i$ and $j$. Suppose that this high-usage link is composed of $m$ trunks of type $H$. Since all virtual circuits between any two network customers homed to AI $i$ and AI $j$ are evenly set up over this high-usage link, the arrival rate for QNA customer type $q$ is $\hat{\lambda}_q = \lambda_p(i, j)/m$. The first queueing station on type $q$'s route is the $P$ station representing AI $i$, and so its service time is characterized by $\hat{\tau}_{q1} = \Delta_A$ and $\hat{c}^2_{sq1} = V\Delta_A/(\Delta_A)^2$. The second, and last, queueing station on types $q$'s route is the $T$ station representing the $n$th high-usage trunk between AI $i$ and AI $j$ and is characterized by $\hat{\tau}_{q2} = s(i, j)/B^H_R$ and $c^2_{sq2} = v_s(i, j)/(B^H_R)^2$.

The squared coefficient of variation for interarrival times for $q$'s route is assigned as $\hat{c}^2_q = v_p(i, j) \cdot \lambda^2_p(i, j)$. This assignment corresponds to the assumption that the variability of traffic arriving at each trunk on a high-usage link is exactly as great as the variability of traffic collectively arriving for transport over the link's entire group of trunks. Such an assumption is not completely accurate. However, if the squared coefficient of variation for interarrival times of the high-usage link traffic is greater than one (i.e., if packet arrivals are burstier than Poisson arrivals, as may typically be the case), then this assumption is a conservative one.

Now consider customer type $q$ corresponding to the flow of traffic from AI $i$ to switch $s$ over the $n$th trunk of the access link from $i$ to $s$. Suppose that this access link is comprised of $m$ trunks of type $H$. The type $q$ customers represent one $m$th of the superposition of all AI $i$ to user $j$ traffic streams for which a path through the network exists via AI $i$ and switch $s$. Therefore, $\hat{\lambda}_q = (1/m)\ \Sigma\lambda_p(i, j)(n(i, j, s)/n(i, j))$ where the summation is over all users $j$ for which a path from $i$ and $j$ involves switch $s$ and where $n(i, j)$ and $n(i, j, s)$ denote, respectively, the number of paths between $i$ and $j$ and the number of paths between $i$ and $j$ via $s$. (Note that $j = i$ is included when AI's do not have any switching capability.) As developed by Albin [34], [35] and Section 4.3 of Whitt [18],

$$\hat{c}^2_q \sim \omega \sum \left[\frac{\lambda^3_p(i, j) \cdot v_p(i, j)}{m \cdot \hat{\lambda}_q} \cdot \frac{n(i, j, s)}{n(i, j)}\right] + 1 - \omega$$

where $\omega$ is the weighting function

$$\omega = \left[1 + 4(1 - \rho_{is})^2 \cdot \left[\left[\sum \left[\frac{\lambda_p(i, j)}{m \cdot \hat{\lambda}_q} \cdot \frac{n(i, j, s)}{n(i, j)}\right]^2\right]^{-1} - 1\right]\right]^{-1},$$

$$\rho_{is} = \frac{\hat{\lambda}_q}{mB^H_R}\left[\frac{\sum \lambda_p(i, j) \cdot s(i, j)}{\hat{\lambda}_q} \cdot \frac{n(i, j, s)}{n(i, j)}\right],$$

and the summations all as above.

The first queueing station on type $q$'s route is the $P$ station representing processing at AI $i$, and so $\hat{\tau}_{q1} = \Delta_A$ and $\hat{c}^2_{sq1} = V\Delta_A/(\Delta_A)^2$. The second and last queueing station is the $T$ station representing the $n$th access link trunk from AI $i$ to switch $s$, and so is characterized by

$$\hat{\tau}_{q2} = \frac{1}{\hat{\lambda}_q} \sum \frac{\lambda_p(i, j)}{m} \cdot \frac{n(i, j, s)}{n(i, j)} \cdot \frac{s(i, j)}{B^H_R}$$

and

$$\hat{c}^2_{sq2} = \left[\frac{1}{[\hat{\lambda}_q(\hat{\tau}_{q2})]^2} \cdot \sum \frac{\lambda_p(i, j)}{m} \cdot \frac{n(i, j, s)}{n(i, j)} \cdot \frac{[v_s(i, j) + s^2(i, j)]}{(B^H_R)^2}\right] - 1.$$

The parameter $\hat{\tau}_{q2}$ is derived from the appropriate averaging of the sizes of all packets which arrive for transport across the access link from AI $i$ to switch $s$, while $\hat{c}^2_{sq2}$ is derived from the application of the fact that the second moment of a mixture of distributions is the mixture of the second moments.

Having determined the input parameters for the QNA customer types, the congestion at each queueing station is analyzed. The QNA approximate congestion measures obtained are, for each station $k$, the mean waiting time $EW_k$, the probability of delay $\sigma_k$, and the squared coefficient of variation of the conditional delay (given delay $> 0$) $c^2_{Dk}$. Equations (44), (48), and (50) of Whitt [18] give the specific derivations of $EW_k$, $\sigma_k$, and $c^2_{Dk}$, respectively. These measures lead to three additional measures for station $k$: the mean conditional delay $ED_k = EW_k/\sigma_k$, the squared coefficient of variation of the waiting time $c^2_{Wk} = [c^2_{Dk} + 1 - \sigma_k]/\sigma_k$, and the waiting time variance $VW_k = c^2_{Wk} \cdot (EW_k)^2$. The squared coefficient of variation of an interdeparture time from station $k$, $c^2_{dk}$, is also calculated and is used in the PNPA algorithm to help characterize the flow of traffic from the access portion into the backbone switching portion of the packet network.

*3) Interswitch Delays:* The delay components for the backbone portion of the network are evaluated by analyzing the section of the decomposed queueing network containing the queueing stations representing processing at the switches and transmission over interswitch links. Fig. 5(d) shows this subnetwork of queues from the entire network given in Fig. 5(b) where three queueing stations have been added in Fig. 5(d). These additional stations, denoted by $X^*$, correspond to artificial holding buffers at switch $X$ for all traffic deliverable to users homed to $X$. In actuality, there is a separate buffer at $X$ for each link from $X$ to a user. However, at this stage in the PNPA network hierarchical decomposition, all of the packets in transport from switch $X$ to users homed to it are aggregated into one traffic stream. Detailed analysis of the delay within switch holding buffers for traffic deliverable to a particular AI or vendor is deferred until after this backbone analysis [see Section IV-B4)], and the use of artificial queueing stations here facilitates that analysis later.

The different QNA customer types in the backbone subnetwork correspond to the aggregate traffic flows between all of the users homed to a particular switch and all of the

users homed to another switch. The first queueing station on each type's route is a $P$ station representing processing at a backbone switch. The last queueing station is an artificial holding buffer at a backbone switch.

As was done in Section IV-B2), the superposition approximations are applied in modeling these aggregate traffic flows. Consider customer type $q$ corresponding to the traffic flow from switch $s_1$ to switch $s_2$ over the $n$th trunk of the $s_1$-to-$s_2$ interswitch link. Suppose that this interswitch link is comprised of $m$ trunks of type $H$. The arrival rate for type $q$ is $\hat{\lambda}_q = (1/m)\,\Sigma\lambda_p(i,j) \cdot (n(i,j,s_1,s_2)/n(i,j))$ where the summation is over all $(i,j)$ pairs for which customer type $q$ includes $i$-to-$j$ traffic. The variability parameter $\hat{c}_q^2$ is determined by $\hat{c}_q^2 = \omega\Sigma\,[(\lambda_p(i,j)/(m \cdot \hat{\lambda}_q)) \cdot n(i,j,s_1,s_2)/n(i,j) \cdot c_{dis}^2] + 1 - \omega$ where

$$\omega = \left[1 + 4(1-\rho_{s_1s_2})^2 \cdot \left[\left[\sum\left[\frac{\lambda_p(i,j)}{m\cdot\hat{\lambda}_q}\cdot\frac{n(i,j,s_1,s_2)}{n(i,j)}\right]^2\right]^{-1} - 1\right]\right]^{-1}$$

and

$$\rho_{s_1s_2} = \frac{\hat{\lambda}_q}{mB_R^H}\left[\sum\frac{\lambda_p(i,j)}{\hat{\lambda}_q}\cdot s(i,j)\cdot\frac{n(i,j,s_1,s_2)}{n(i,j)}\right] \cdot \frac{\lambda_p(i,j)\cdot s(i,j)}{\hat{\lambda}_q}\cdot\frac{n(i,j,s_1,s_2)}{n(i,j)}.$$

For this customer type, there are four queueing stations along its route. The first and third are the $P$ stations for processing at switches $s_1$ and $s_2$. Their service times are characterized by $\hat{\tau}_{q1} = \hat{\tau}_{q3} = \Delta_S$ and $\hat{c}_{sq1}^2 = \hat{c}_{sq3}^2 = V\Delta_S/(\Delta_S)^2$. The second station is the $T$ station representing the $n$th $s_1$-to-$s_2$ interswitch trunk. Its service time parameters are derived from the approximate mixing of the mean and second moment of packet sizes over all the user $i$ to user $j$ traffic streams included in the aggregate type $q$ traffic flow. Specifically,

$$\hat{\tau}_{q2} = \frac{1}{\hat{\lambda}_q}\sum\frac{\lambda_p(i,j)}{m}\cdot\frac{n(i,j,s_1,s_2)}{n(i,j)}\cdot\frac{s(i,j)}{B_R^H}$$

and

$$\hat{c}_{sq2}^2 = \left[\frac{1}{[\hat{\lambda}_q(\hat{\tau}_{q2})]^2}\cdot\sum\frac{\lambda_p(i,j)}{m}\cdot\frac{n(i,j,s_1,s_2)}{n(i,j)}\cdot\frac{[v_s(i,j)+s^2(i,j)]}{(B_R^H)^2}\right] - 1.$$

The fourth station is the artificial holding buffer at switch $s_1$ and its service time parameters $\hat{\tau}_{q4}$ and $\hat{c}_{sq4}^2$ are set to 0 and 1, respectively, which ensures that zero delay will be assessed to the artificial queueing stations in this queueing network analysis.

Similarly, consider customer type $q$ corresponding to the flow of traffic among users homed to the same switch $s$. The customer arrival rate is $\hat{\lambda}_q = \Sigma\lambda_p(i,j)(n(i,j,s)/n(i,j))$, the arrival variability parameter is

$$\hat{c}_q^2 = \omega\sum\left[\frac{\lambda_p^3(i,j)\cdot v_p(i,j)}{\hat{\lambda}_q}\cdot\frac{n(i,j,s)}{n(i,j)}\right] + 1 - \omega$$

where

$$\omega = \left[1 + 4(1-\rho_s)^2 \cdot \left[\left[\sum\left[\frac{\lambda_p(i,j)}{\hat{\lambda}_q}\cdot\frac{n(i,j,s)}{n(i,j)}\right]^2\right]^{-1} - 1\right]\right]^{-1}$$

and $\rho_s = \hat{\lambda}_q/\Delta_s$. The two queueing stations along type $q$'s route are first, the $P$ station at $s$ and second, the artificial buffer at $s$ with service times characterized by $\hat{\tau}_{q1} = \Delta_S$, $\hat{c}_{sq1}^2 = V\Delta_S/(\Delta_S)^2$, $\hat{\tau}_{q2} = 0$, and $\hat{c}_{sq2}^2 = 1$.

From all of the above inputs for customer types, the QNA approximate congestion measures obtained from each queueing station $k$ include $EW_k$, $VW_k$, and $c_{Wk}^2$, the mean, variance, and squared coefficient for the waiting time; $\sigma_k$, the probability of delay; $ED_k$, $VD_k$, and $c_{Dk}^2$, the mean, variance, and squared coefficient of variation for the conditional delay; and $EN_k$ and $VN_k$, the mean and variance for the number of customers present (including any in service). For each queueing station $k$ corresponding to artificial buffers at the switches $c_{dk}^2$, the squared coefficient of variation of interdeparture times is also obtained. The $c_{dk}^2$'s are later used to help characterize the flow of traffic from the network's interswitch portion out to the access portion.

*4) Switch to User Delays:* The delay components corresponding to the transport of traffic from the backbone switches to the individual AI's and vendors homed to them are evaluated by analyzing the section of the decomposed network of queues containing $T$ stations from switches to AI's and from switches to vendors, and $P$ stations at the AI's. Fig. 5(e) shows this subnetwork for the network in Fig. 5(b).

Consider QNA customer type $q$ in this subnetwork of queues corresponding to the flow of traffic from switch $s$ to AI $i$ over the $n$th trunk of the access link between $s$ and $i$. Suppose that this access link is comprised of $m$ type $H$ trunks. Hence, $\hat{\lambda}_q = (1/m)\,\Sigma\lambda_p(j,i) \cdot (n(i,j,s)/n(i,j))$. These type $q$ customers are part of the traffic passing through the artificial buffering station at $s$ in the backbone [Section IV-B3)] subnetwork. Let $c_{dk(s^*)}^2$ denote the previously QNA-determined squared coefficient of variation of interdepartures from this artificial buffer and $\lambda_{k(s^*)}$ the overall arrival rate to the artificial buffer. The variability parameter for type $q$ customers here is taken to be $\hat{c}_q^2 = (\hat{\lambda}_q/\lambda_{k(s^*)}) \cdot c_{dk(s^*)}^2 + [1 - (\hat{\lambda}_q/\lambda_{k(s^*)})]$.

This characterization follows from the approximation that the departure process from queueing station $k(s^*)$ is a renewal process and the fact that a renewal split by independent probabilities is again a renewal process.

The first queueing station on type $q$'s route is the $T$ station representing the $n$th trunk from switch $s$ to AI $i$. Its service time distribution has mean

$$\hat{\tau}_{q1} = \frac{1}{\hat{\lambda}_q} \sum \frac{\lambda_p(j,i)}{m} \cdot \frac{n(i,j,s)}{n(i,j)} \cdot \frac{s(j,i)}{B_R^H}$$

and variability parameter

$$\hat{c}_{sq1}^2 = \left[ \frac{1}{\hat{\lambda}_q^2} \cdot \sum \frac{\lambda_p(j,i)}{m} \cdot \frac{n(i,j,s)}{n(i,j)} \cdot \frac{[v_s(j,i) + s^2(j,i)]}{(B_R^H)^2} \right] - 1.$$

The second, and last, queueing station on type $q$'s route is the $P$ station for AI $i$, and so $\hat{\tau}_{q2} = \Delta_A$ and $\hat{c}_{sq2}^2 = V\Delta_A/(\Delta_A)^2$. (For customer type $q$ corresponding to the flow of traffic from switch $s$ to vendor $i$ over the $n$th trunk of the access link, there is only one queueing station on the route, namely, the corresponding $T$ station.)

*5) User-to-User Delay Distributions:* A distribution $F_{ij}^p(t)$ of $i$-to-$j$ packet transport time over a particular path $p$ is now constructed, for all users $i$ and $j$ and paths connecting them, from the individual delay components analyzed in Sections IV-B2)–IV-B4). The mean and variance $i$-to-$j$ packet transport time along $p$ and the mean and variance of this transport time conditioned on its exceeding the sum of the mean PT and TD components along $p$ are first calculated from the various queueing station performance measures previously determined. Then $F_{ij}^p(t)$ is constructed so as to match these four $i$-to-$j$ delay characteristics. We discuss below this procedure for users connected by a two-switch path. Cases for users connected by a high-usage link path or one-switch path are similarly handled; those cases just involve fewer individual delay components along the paths.

*Two-Switch Paths:* Consider users $i$ and $j$ connected by path $p$ involving two packet switches $s_1$ and $s_2$, with user $i$ homed to $s_1$ and user $j$ homed to $s_2$. In this case, the total packet transport time for $i$-to-$j$ is the *sum* of PD and PT at AI $i$, QD and TD for the $i$-to-$s_1$ access link, PD and PT at $s_1$, QD and TD for the $s_1$-to-$s_2$ interswitch link, PD and PT at $s_2$, and QD and TD for the $s_2$-to-$j$ access link. The mean $i$-to-$j$ packet transport time along $p$ is therefore

$$\begin{aligned} ET_{\text{pkt}}^p(i,j) = {} & \left[ \Delta_A + EW_i + EW_{is_1} + \frac{s(i,j)}{B_R^{H_A}} \right] \cdot 1_{\{i \text{ is an AI}\}} \\ & + \frac{s(i,j)}{B_R^{H_V}} \cdot 1_{\{i \text{ is a VN}\}} + 2\Delta_s + EW_{s_1} \\ & + EW_{s_1 s_2} + \frac{s(i,j)}{B_R^{H_S}} + EW_{s_2} + EW_{s_2 j} \\ & + \left[ \Delta_A + EW_j + \frac{s(i,j)}{B_R^{H_A}} \right] \cdot 1_{\{j \text{ is an AI}\}} \\ & + \frac{s(i,j)}{B_R^{H_V}} \cdot 1_{\{j \text{ is a VN}\}} \end{aligned} \tag{4.1}$$

where $1_{\{\ \}}$ is the indicator function and where $H_A$, $H_V$, and $H_s$ are, respectively, the type of trunks connecting AI's to switches, vendors to switches, and between switches. The terms $EW_i$ and $EW_j$ in (4.1) denote the mean waiting times as calculated in Section IV-B2) for the $P$ stations representing AI's $i$ and $j$, and $EW_{s1}$ and $EW_{s2}$, the mean waiting times for the $P$ stations representing switches $s_1$ and $s_2$ [Section IV-B3)]. Thus, $EW_i$, $EW_j$, $EW_{s1}$, and $EW_{s2}$ are the four mean PD components along $p$. The terms $EW_{is1}$, $EW_{s1s2}$, and $EW_{s2j}$ denote the mean waiting times for the $T$ stations representing one of the trunks on the $i$-to-$s_1$ access link, $s_1$-to-$s_2$ interswitch link, and $s_2$-to-$j$ access link, respectively, as calculated through the analysis of Sections IV-B2)–IV-B4). These three terms are the mean QD components along $p$. The mean PT components are given by the $\Delta_A$ and $\Delta_S$ terms, and the mean TD components along $p$ are given by the $s(i,j)$-divided-by-$B_R$ terms. Note that we use the ratios of $i$-to-$j$ mean packet size and transport speeds as the mean TD terms instead of the global mean TD figures for $i$-to-$s_1$, $s_1$-to-$s_2$, and $s_2$-to-$j$ as averaged over all the packet sizes carried by those links. In adopting this approach, we have adjusted the previously determined queueing station equilibrium performance measures to reflect the specific characteristics of $i$-to-$j$ traffic.

The variance of $i$-to-$j$ packet transport time along $p$ is approximated by

$$\begin{aligned} VT_{\text{pkt}}^p(i,j) = {} & \left[ V\Delta_A + VW_i + VW_{is_i} + \frac{v_s(i,j)}{(B_R^{H_A})^2} \right] \cdot 1_{\{i \text{ is an AI}\}} \\ & + \frac{v_s(i,j)}{(B_R^{H_V})^2} \cdot 1_{\{i \text{ is a VN}\}} + 2V\Delta_S + VW_{s_1} \\ & + VW_{s_1 s_2} + \frac{v_s(i,j)}{(B_R^{H_S})^2} + VW_{s_2} + VW_{s_2 j} \\ & + \left[ V\Delta_A + VW_j + \frac{v_s(i,j)}{(B_R^{H_A})^2} \right] \\ & \cdot 1_{\{j \text{ is an AI}\}} + \frac{v_s(i,j)}{(B_R^{H_V})^2} \cdot 1_{\{j \text{ is a VN}\}} \end{aligned} \tag{4.2}$$

where $VW_i$, $VW_{s_1}$, $VW_{s_2}$, and $VW_j$ are the variances of waiting times at the appropriate $P$ stations, and $VW_{is_1}$, $VW_{s_1s_2}$, and $VW_{s_2}j$ are the waiting time variances at the appropriate $T$ stations. These then are the variances of the PD and QD components along $p$. The variances of the TD components are the $v_s(i,j)$-divided-by-$(B_R)^2$ terms in (4.2).

The sum of the mean PT and TD components along $p$ is

$$\begin{aligned} & \left[ \Delta_A + \frac{s(i,j)}{B_R^{H_A}} \right] [1_{\{i \text{ is a VN}\}} + 1_{\{j \text{ is a VN}\}}] \\ & + \frac{s(i,j)}{B_R^{H_V}} + [1_{\{i \text{ is a VN}\}} + \cdot\ 1_{\{j \text{ is a VN}\}}] \\ & + 2\Delta_S + \frac{s(i,j)}{B_R^{H_S}}, \end{aligned} \tag{4.3}$$

and the probability that the $i$-to-$j$ packet transport time exceeds (4.3) is approximated by $\bar{\sigma}_{ij}^{p} = 1 - (1 - \sigma_i \cdot 1_{\{i \text{ is an AI}\}}) \cdot (1 - \sigma_{is_1} \cdot 1_{\{i \text{ is an AI}\}})(1 - \sigma_{s_1}) \cdot (1 - \sigma_{s_1 s_2})(1 - \sigma_{s_2})(1 - \sigma_{s_2 j}) \cdot (1 - \sigma_j \cdot 1_{\{j \text{ is an AI}\}})$ where $\sigma_i$, $\sigma_j$, $\sigma_{s_1}$, and $\sigma_{s_2}$ are the probabilities of delay at the $P$ stations representing AI's $i$ and $j$ and switches $s_1$ and $s_2$. The terms $\sigma_{is_1}$, $\sigma_{s_1 s_2}$, and $\sigma_{s_2 j}$ are the probabilities of delay at the $T$ stations representing the $i$-to-$s_1$, $s_1$-to-$s_2$, and $s_2$-to-$j$ links. The mean and variance of $i$-to-$j$ packet transport time along $p$ conditioned on its exceeding (4.3) are thus approximated by

$$\left[\Delta_A + \frac{s(i,j)}{B_R^{H_A}}\right] \cdot [1_{\{i \text{ is an AI}\}} + 1_{\{j \text{ is an AI}\}}] + \frac{1}{\bar{\sigma}_{ij}^{p}} [EW_i + EW_{is_1}] \cdot 1_{\{i \text{ is an AI}\}} + 2\Delta_S + \frac{s(i,j)}{B_R^{H_V}} + \frac{1}{\bar{\sigma}_{ij}^{p}} [EW_{s_1} + EW_{s_1 s_2} + EW_{s_2} + EW_{s_2 j}] + \frac{s(i,j)}{B_R^{H_V}} \cdot [1_{\{i \text{ is an AI}\}} + 1_{\{j \text{ is an AI}\}}] \text{ and } \cdot \frac{1}{\bar{\sigma}_{ij}^{p}} [VW_{s_1} + VW_{s_1 s_2} + VW_{s_2} + VW_{s_2 j} + [VW_i + VW_{is_1}] \cdot 1_{\{i \text{ is an AI}\}}], \text{ respectively.} \quad (4.5)$$

The distribution $F_{ij}^{p}(t)$ is now chosen to have an atom at (4.3) and a density above (4.3) which matches $ET_{\text{pkt}}^{p}(i, j)$, $VT_{\text{pkt}}^{p}(i, j)$, and (4.3)–(4.5) above. The PNPA algorithm uses a hyperexponential density with balanced means (see (55) in [16]) if the squared coefficient of variation of the conditional $i$-to-$j$ packet transport time $\bar{c}_{ij}^2$ [the quantity in (4.5) divided by the square of the quantity in (4.4)] is greater than one. If $0.99 \leq \bar{c}_{ij}^2 \leq 1$, then an exponential density with mean equal to (4.1) is used; if $0.5 \leq \bar{c}_{ij}^2 < 0.99$, then a convolution of two exponential distributions is used; and if $\bar{c}_{ij}^2 < 0.5$, then an Erlang distribution with shape parameter 2 and scale parameter equal to two times the reciprocal of (4.1) is used.

*Network Delay Distribution:* As a result of the network routing rules, the distribution $F_{ij}(t)$ of network delay from user $i$ to user to determined by the average $F_{ij}(t) = \Sigma F_{ij}^{p}(t)/n(i, j)$ where the summation is over all the $n(i, j)$ $i$-to-$j$ paths. We use $ET_{\text{pkt}}(i, j)$ and $VT_{\text{pkt}}(i, j)$ to denote the mean and variance of $F_{ij}(t)$. Since the distribution of message transport time from $i$ to user $j$ for an originating customer of service class $c$ is determined by $F_{ijc}(t) = F_{ij}(t - \text{AT}_c)$, its mean and variance are $\text{AT}_c + ET_{\text{pkt}}(i, j)$, and $VT_{\text{pkt}}(i, j)$, respectively. These are checked against the message transport time requirements $\delta_c$ and $\gamma_c$. In identifying performance bottlenecks, the various individual delay component performance measures are of interest as the mean utilizations of the network's AI's, switches, and transmission links. The utilization figures are the traffic intensities $\rho_s$ calculated for each queueing station during the various queueing network analyses. All of the PNPA information concerning performance bottlenecks is then used to update the equipment derating factors used by the OPND module.

*a) Retransmissions:* The actual load (in packets) carried by each AI, switch, and transmission link in the network includes those packets to be transported as a result of the retransmission of a message. As discussed in Section II-D, the PNPA algorithm considers message retransmission conducted on an edge-to-edge basis which is due to errors in transmission over network links or excessive delay in transport. In order to analyze the effects of retransmissions on the network's message transport time performance measures, the PNPA algorithm augments the equilibrium analysis of Section IV-B. The steps involved are to evaluate the percentage of traffic that is retransmitted from user $i$ to user $j$, to update the carried load from $i$ to $j$ to include the added load induced by this retransmitted traffic, and to adjust the $F_{ij}(t)$ packet transport time distribution accordingly for every pair of users $i$ and $j$.

In the absence of transmission errors across the network, any retransmission of a message for an originating customer of service class $c$ is due to the expiration of the time-out threshold $T_c$. The end-to-end message transport time distribution $F_{ijc}(t)$ evaluated at the time-out threshold gives an indication of whether the $i$-to-$j$ message transport time remains within the threshold when the packet network is loaded at the original offered traffic levels. Therefore, $P_{\text{ret}}(i, j, c)$, the proportion of messages sent from $i$ to $j$ by the customer class $c$ which are retransmitted, is first broadly estimated by $P_{\text{ret}}(i, j, c) = 1 - F_{ijc}(T_c)$.

To account for transmission errors, we determine the probability $P_{\text{err}}(i, j, c)$ of a transmission error anywhere in transport for an $i$-to-$j$ message originated by a class $c$ customer. This $P_{\text{err}}(i, j, c)$ is the average of the probabilities of a transmission error over all paths connecting user $i$ and user $j$. The probability of a transmission error over a specific path $p$, $P_{\text{err}}^{p}(i, j, c)$, is one minus the probability that there is no error along each of the transmission links on the path. For the access loop, the probability of no transmission error for a class $c$ message is simply computed as $\eta_c = (1 - \upsilon_c)^{Z_c}$ where $\upsilon_c$ is the loop's BER and $Z_c$ is the mean class $c$ message size. For each network link $L$ comprised of type $H$ trunks, the probability of no error in the transmission of one packet over $L$ is $\eta_L = (1 - \upsilon_H)^{s(i,j)}$. In computing $\eta_c$ and $\eta_L$, it is assumed that: 1) bit errors on any link are i.i.d. with the probability of any one bit being in error equal to that link's BER, and 2) a transmitted message or packet is considered to be in error if any one bit of the message or packet is in error. These assumptions warrant further attention. If instead of the first assumption, the occurrence of bit errors over individual network links are correlated in a dependent fashion, then calculations of $\eta_c$ and $\eta_L$ based on bursty error models should be used. (Such bursty error models typically result in higher values for $\eta_c$ and $\eta_L$.) For some packet network applications such as in packetized voice communications, the second assumption may also be too rigid, in which case $\eta_c$ and $\eta_L$ should be corrected to allow for some

threshold number of errored bits within a packet or a message.

Relevant link BER's for present packet networks are no larger than $10^{-3}$ and are typically on the order of $10^{-5}$ or $10^{-6}$. The size of packets is usually greater than 100 bits, sometimes as large as 2048 bits, and the size of messages is even greater. Therefore, in the numerical computation of $\eta_c$ and $\eta_L$, we use the exponential approximation $e^{-ny}$ for $(1 - y)^n$ so as to minimize computational effort and errors. We have found that for the relevant ranges of BER, packet size, and message size, this approximation retains accuracy to at least the fourth decimal place.

The probability of a transmission error in the transport of a customer class $c$ message from $i$ to $j$ over path $p$ is then $P^p_{\text{err}}(i, j, c) = 1 - \eta_c \cdot \Pi^l_{k=1} \eta_{L_k}^{z_c/s(i,j)}$ where $L_1, L_2, \cdots, L_l$ are the network transmission links along $p$. Given $P^p_{\text{err}}(i, j, c)$ for every $i$-to-$j$ path $p$, the overall transmission error rate is $P_{\text{err}}(i, j, c) = \Sigma P^p_{\text{err}}(i, j, c)/n(i, j)$ where the summation is over all the $i$-to-$j$ paths. Our estimation of $P_{\text{ret}}(i, j, c)$ is thus extended to $P_{\text{ret}}(i, j, c) = 1 - F_{ijc}(T_c)$ $[1 - P_{\text{err}}(i, j, c)]$.

Having evaluated $P_{\text{ret}}(i, j, c)$ for all users $i$ and $j$ and all customer classes $c$, the PNPA algorithm updates the equilibrium transport time analysis to reflect the added load in packets on the network induced by retransmitted traffic. This is accomplished by increasing the point-to-point packet throughput levels $\lambda_p(i, j)$ to include retransmitted traffic, and then reassessing the packet transport time distributions by returning to the analyses found in Sections IV-B2)–IV-B5). In particular, we increase $\lambda_p(i, j)$ to $\lambda'_p(i, j) = \lambda_p(i, j)\ [1 + \Sigma_c\ \theta_c P_{\text{ret}}(i, j, c)]$. Summing over all user-to-user pairs, the quantity $\Omega$ defined by $\Omega = \Sigma_i \Sigma_j$ $[\lambda_p(i, j)\ \Sigma_c\ \theta_c P_{\text{ret}}(i, j, c)]/\Sigma_i \Sigma_j \lambda'_p(i, j)$ gives a total measure of the proportion of network carried traffic which is retransmitted traffic.

Each repetition of the above retransmission/transport time analysis only macroscopically approximates the effects of retransmissions on network performance. This is because all of the end-to-end analyses presented in this paper deal with an equilibrium characterization of performance and do not incorporate consideration of the detailed, dynamic mechanisms implemented by flow control schemes. However, this limited iterative analysis should be sufficient for packet network design purposes since flow control schemes do not generally permit traffic to be retransmitted indefinitely. We suggest carrying out two cycles of transmission/transport time analysis. An indication of substantial retransmitted traffic on either the first or second iteration (e.g., $\Omega \geq 0.3$) can be interpreted as evidence that the network is too heavily loaded. In such a case, it is appropriate to stimulate the OPND module of the PANDA methodology to reconfigure the packet network even if all of the network performance requirements had been satisfied.

## V. Concluding Remarks

We have described realistic models and practical solution procedures for the design and performance analysis of backbone packet switching networks. These have been incorporated into a software package by the authors (while members of AT&T Bell Laboratories) which is currently being used by various organizations within AT&T Bell Laboratories, AT&T Information Systems, AT&T Technologies, and Bell Communications Research to investigate issues in the design and performance of packet networks. The OPND and PNPA modules, described in Sections III and IV, have been found to be very efficient; a problem with seven switches, nine vendors, and 52 AI's requires less than 1 min of CPU time on an IBM/Amdahl computer. One design and performance analysis of a network involving 15 packet switches generally requires less than 3 min.

Software simulations of small networks and lab measurements have been used to initially validate the PNPA delay analysis and to fine tune PNPA parameters. For certain nonpriority switching technologies, the results have been encouraging with respect to accuracy suitable for engineering purposes. We have not found it possible, however, to efficiently capture in simulations the network effects on performance for networks with greater than three switches and for which protocol details, such as retransmission mechanisms, are incorporated. Therefore, what is needed is direct comparison of PNPA results against end-to-end performance measurements from actual packet networks applications as these measurements become available.

## Appendix
## Summary of Notation

*Technology Parameters*

$C_A$ = total number of virtual circuits simultaneously supportable by an access interface.

$C_S$ = total number of virtual circuits simultaneously supportable by a switch.

$C_R^H$ = total number of virtual circuits simultaneously supportable by one transmission facility of the $H$th type.

$B_R^H$ = transport speed (in bits per second) for one transmission facility of the $H$th type.

$\Delta_A$ = mean processing time (in seconds) for an access interface.

$V\Delta_A$ = variance of the processing time for an access interface.

$\Delta_S$ = mean processing time (in seconds) for a switch.

$V\Delta_S$ = variance of the processing time for a switch.

$\upsilon_H$ = BER for one transmission facility of the $H$th type.

$A_N^H$ = mean number of call attempts per second supportable by an NPU of type $H$.

$A_S$ = mean number of call attempts per second supportable by a switch.

$P_N^H$ = mean number of packets per second supportable by an NPU of type $H$.

$P_S$ = mean number of packets per second supportable by a switch.

$R_N^H$ = total number of trunks supportable by an NPU of type $H$.
$N_S$ = total number of NPU's supportable by a switch.
$G_A$ = fixed cost for an access interface.
$F_S$ = fixed cost for a switch.
$G_T^H$ = fixed cost for an AI NPU of type $H$.
$F_T^H$ = fixed cost for a switch NPU of type $H$.
$F_R^H$ = cost per mile for a trunk of type $H$.
$O_i$ = homing option for user $i$, i.e., "one" means home to one switch, "all" means home to all switches, "subset" means home to any subset of switches.

*User i-to-User j Traffic*

$\lambda_c(i, j)$ = mean arrival rate of requests for call setup from $i$ to $j$ (in calls per sound).
$\mu_c^{-1}(i, j)$ = mean holding time of calls setup from $i$ to $j$.
$\lambda_p(i, j)$ = mean arrival rate of packets for transport from $i$ to $j$ (in packets per second).
$v_p(i, j)$ = variance of interarrival times of packets for transport from $i$ to $j$.
$s(i, j)$ = mean size of packets in transport from $i$ to $j$ (in bits).
$v_s(i, j)$ = variance of size of packets in transport from $i$ to $j$.
$S(i, j)$ = mean size of messages in transport from $i$ to $j$ (in packets per message).
$V_S(i, j)$ = variance of size of messages in transport from $i$ to $j$.
$n(i, j)$ = total number of least resistance paths between $i$ and $j$.
$n(i, j, s)$ = number of least resistance paths via switch $s$ between $i$ and $j$.
$n(i, j, s_1, s_2)$ = number of least resistance paths via switches $s_1$ and $s_2$ between $i$ and $j$.

*Network Utilization*

$U_p(i, s)$ = mean number of packets per second in the direction from user $i$ to switch $s$; $U_p(s, i)$ represents the traffic in the other direction.
$U_p(s_1, s_2)$ = mean number of packets per second in the direction from switch $s_1$ to $s_2$.
$U_p(s)$ = mean number of packets per second at switch $s$.
$U_A(i, s)$ = mean number of call attempts per second in the direction from user $i$ to switch $s$.
$U_A(s_1, s_2)$ = mean number of call attempts per second in the direction from switch $s_1$ to $s_2$.
$U_A(s)$ = mean number of call attempts per second at switch $s$.

*Network Design*

$L_s$ = users clustered to the switch $s$.
$S$ = number of switches.
$N_s^H$ = number of NPU's of type $H$ at switch $s$.
$R_{is}$ = number of trunks between user $i$ and switch $s$.
$R_{s_1s_2}$ = number of trunks between switch $s_1$ and switch $s_2$.
$H_A$ = type of trunk used for access interfaces to switches.
$H_v$ = type of trunk used between vendors and switches.
$H_S$ = type of trunk used between switches.

*Design Parameters*

$f_{AS}$ = derating factor for switch call attempts per second capacity.
$f_{AN}^H$ = derating factor for NPU call attempts per second capacity of $H$th type.
$f_{PS}$ = derating factor for switch packet per second capacity.
$f_{PN}^H$ = derating factor for NPU packet per second capacity of the $H$th type.
$f_{NS}^H$ = derating factor for switch capacity in number of NPU's.
$f_{RN}^H$ = derating factor for NPU capacity in number of trunks of the $H$th type.
$f_{BR}^H$ = derating factor for transport speed for one transmission facility of the $H$th type.
$f_{CR}^H$ = derating factor for number of virtual circuits simultaneously supportable by one transmission facility of type $H$.

*Network Customer Service Classes*

$L_c$ = access speed (in bits per second) for a customer of the $c$th customer service class.
$v_c$ = BER for access loop of the $c$th customer service class.
$Z_c$ = mean size of messages for the $c$th customer service class (in bits per message).
$T_c$ = time-out threshold in message transport time for the $c$th customer service class.
$\Theta_c$ = proportion of total network customers belonging to the $c$th customer service class.
$\beta$ = network (mean) blocking requirement for all end-to-end pairs.
$\delta_c$ = mean message transport time requirement between all end-to-end pairs for the $c$th customer service class.
$\gamma_c$ = variance of message transport time requirement between all end-to-end pairs for the $c$th customer service class.

*Blocking Performance Measures*

$b(i, j)$ = end-to-end blocking level between users $i$ and $j$.
$b_{ij}^p(N_1, \cdots, N_n, L_1, \cdots, L_l)$ = end-to-end blocking between users $i$ and $j$ over the path $p$ involving processing nodes $N_1, \cdots, N_n$ and transmission links $L_1, \cdots, L_l$.

*Transport Time Performance Measures*

$F_{ijc}(t)$ = probability distribution of message transport time from user $i$ to user $j$ for an originating customer of service class $c$.

$F_{ij}(t)$ = probability distribution of packet transport time from user $i$ to user $j$.

$F^{p}_{ij}(t)$ = probability distribution of packet transport time from user $i$ to user $j$ over a particular path $p$.

$ET_{\text{pkt}}(i, j)$ = mean packet transport time $i$-to-$j$.

$VT_{\text{pkt}}(i, j)$ = variance of packet transport times $i$-to-$j$.

$ET^{p}_{\text{pkt}}(i, j)$ = mean packet transport time $i$-to-$j$ over path $p$.

$VT^{p}_{\text{pkt}}(i, j)$ = variance of packet transport times $i$-to-$j$ over path $p$.

*Retransmission Performance Measures*

$P_{\text{ret}}(i, j, c)$ = probability that a message from user $i$ to user $j$ for an originating customer of service class $c$ is retransmitted.

$P_{\text{err}}(i, j, c)$ = probability of a transmission error anywhere in transport for an $i$-to-$j$ message originated by a class $c$ customer.

$P^{p}_{\text{err}}(i, j, c)$ = probability of a transmission error anywhere in transport over a particular path $p$ for an $i$-to-$j$ message originated by a class $c$ customer.

$\Omega$ = proportion of total network carried traffic which is retransmitted traffic.

## Acknowledgment

Many people have aided us in this work, primarily through discussions on data communications technologies, network optimization issues, and performance analysis theory, and through the development and testing of the PANDA computer code. We would especially like to thank E. M. Arkin of Stanford University; S. K. Martens, W. Pehlert, K. Raimer, P. Schuhmann, A. T. Seery, R. A. Stubing, and W. Whitt of AT&T Bell Laboratories; A. E. Magnus and C. F. Newman of Bell Communications Research; and G. E. Herman of AT&T-IS. At some point during the course of our work on PANDA, all of the above were at AT&T Bell Laboratories.

## References

[1] M. Schwartz, *Computer-Communication Network Design and Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[2] A. S. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[3] H. Frank and W. Chou, "Topological optimization of computer networks," *Proc. IEEE*, vol. 60, pp. 1385–1397, Nov. 1972.

[4] R. R. Boorstyn and H. Frank, "Large-scale network topological optimization," *IEEE Trans. Commun.*, vol. COM-25, pp. 29–47, Jan. 1977.

[5] M. Gerla and L. Kleinrock, "Topological design of distributed computer networks," *IEEE Trans. Commun.*, vol. COM-25, pp. 48–60, Jan. 1977.

[6] J. M. McQuillan and D. C. Walden, "The ARPA network design decisions," *Comput. Networks*, vol. 1, pp. 243–289, Aug. 1977.

[7] P. J. Kuen, "Analysis of switching system control structures by decomposition," in *Congressbook*, 9th ITC, 1977.

[8] K. Sriram and W. Whitt, "Characterizing superposition arrival processes and the performance of multiplexers for voice and data," in *Proc. GLOBECOM'85*, 1985, p. 25.4.

[9] P. McGregor and D. Shen, "Network design: An algorithm for the access facility location problem," *IEEE Trans. Commun.*, vol. COM-25, pp. 61–73, Jan. 1977.

[10] A. Kershenbaum and R. Boorstyn, "Centralized teleprocessing network design," in *Proc. NTC'75*, Dec. 1975, pp. 27.11–27.14.

[11] E. Feldman, F. A. Lehner, and T. L. Ray, "Warehouse location under continuous economies of scale," *Management Sci.*, vol. 12, pp. 670–684, May 1966.

[12] A. Kershenbaum and W. Chou, "A unified algorithm for designing multidrop teleprocessing networks," *IEEE Trans. Commun.*, vol. COM-22, pp. 1762–1772, Nov. 1974.

[13] J. F. Hayes, "An adaptive technique for local distribution," *IEEE Trans. Commun.*, vol. COM-26, pp. 1178–1186, Aug. 1978.

[14] W. Chou, F. Ferrante, M. Balagangadhar, and L. Gerke, "An integrated approach to optimally locating network access facilities," in *Proc. ICC'78*, 1978, pp. 335–345.

[15] D. D. Sheng, "Performance analysis methodology for packet network design," in *Proc. GLOBECOM'83*, 1983, p. 13.5.1.

[16] R. B. Cooper, *Introduction to Queueing Theory*, 2nd ed. New York: North-Holland, 1981.

[17] D. Jagerman, "Some properties of the Erlang loss function," *Bell. Syst. Tech. J.*, vol. 53, pp. 525–551, Mar. 1974.

[18] W. Whitt, "The queueing network analyzer," *Bell. Syst. Tech. J.*, vol. 62, pp. 2779–2815, Nov. 1983.

[19] W. Kraemer and M. Langenbach-Belz, "Approximate formulae for the delay in the queueing system *GI/G/1*," in *Proc. 8th Int. Teletraffic Congr.*, Melbourne, Australia, 1976, p. 235-1/8.

[20] G. L. Chesson and A. G. Fraser, "Datakit network architecture," in *Proc. COMPCON*, Spring 1980, pp. 59–61.

[21] J. C. Ehlinger and R. W. Stubblefield, "No. 1PSS: Number 1 packet switching system service capabilities and architecture," in *Proc. ICC'83*, 1983, p. E6.1.1.

[22] A. L. Dudick, E. Fuchs, and P. E. Jackson, "Data traffic measurements for inquiry-response computer communication systems," in *Proc. IFIP*, Ljubljana, Yogoslavia, Aug. 1971, pp. 634–641; reprinted in *Advances in Computer Communications and Networking*, W. W. Chu, Ed. Dedham, MA: Artech.

[23] E. Fuchs and P. E. Jackson, "Estimates of distributions of random variables for certain computer communications traffic models," *Commun. ACM*, vol. 13, no. 12, pp. 752–757, 1970; reprinted in W. W. Chu, Ed., *Advances in Computer Communications and Networking*. Dedham, MA: Artech.

[24] R. B. Gamble, H. R. Seltzer, M. Speter, and M. Westheiner, "30/20 GHz fixed communications systems service demand assessment," NASA Lewis Res. Cen., Cleveland, OH, NASA Rep. CR 159620, Aug. 1979.

[25] P. F. Pawlita, "Traffic measurements in data networks: Recent measurement results, and some implementations," *IEEE Trans. Commun.*, vol. COM-29, pp. 525–535, 1981.

[26] P. F. Pawlita and H. D. Suedhofen, "User behavior in teleprocessing networks: Analytical models based on empirical data," in *Proc. 10th Int. Teletraffic Congr.*, Montreal, P.Q., Canada, 1983, p. 3.3.4.

[27] W. Whitt, "Blocking when service is required from several facilities simultaneously," *AT&T Tech. J.*, to be published.

[28] S. Halfin, "Batch delays versus customer delays," *Bell Syst. Tech. J.*, vol. 62, pp. 2011–2015, Sept. 1983.

[29] W. Whitt, "Comparing batch delays and customer delays," *Bell. Syst. Tech. J.*, vol. 62, pp. 2001–2009, Sept. 1983.

[30] ——, "Performance of the queueing network analyzer," *Bell. Syst. Tech. J.*, vol. 62, pp. 2817–2843, Nov. 1983.

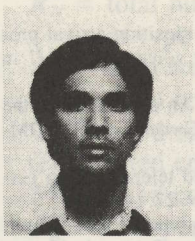
[31] P. J. Kuehn, "Approximate analysis of general queuing networks by decomposition," *IEEE Trans. Commun.*, vol. COM-25, pp. 61–73, 1977.

[32] M. Butto, G. Colombo, and A. Tonietti, "Packet network performance analysis," *CSELT Rapporti Fechici*, vol. IX, no. 1, pp. 45–58.

[33] E. Gelenbe and I. Mitrani, *Analysis and Synthesis of Computer Systems*. New York: Academic, 1980.

[34] S. A. Albin, "Approximating queues with superposition arrival processes," Ph.D dissertation, Dep. Industrial Eng. Oper. Res., Columbia Univ., New York, NY, 1981.

[35] ——, "Approximating a point process by a renewal process, II: Superposition arrival processes to queues," Dep. Industrial Eng., Rutgers Univ., New Brunswick, NJ, 1982.



**Clyde L. Monma** received the B.S. degree in computer science from Washington State University, Pullman, WA, in 1974, and the M.S. and Ph.D. degrees in operations research from Cornell University, Ithaca, NY, in 1976 and 1978, respectively.

He joined the Operations Research Center at Bell Laboratories in 1979, and transferred to the Mathematical, Communications, and Computer Sciences Laboratory at the formation of Bell Communications Research, Morristown, NJ, in 1983. He has over 20 published papers in the areas of discrete mathematics, combinatorial optimization, and operations research, with a particular interest in network design problems, scheduling, and algorithm development.

Dr. Monma is very active in the professional community as a Co-Managing Editor of the *Journal of Graph Theory*, Associate Editor of *Management Science*, *Naval Research Logistics Quarterly*, *Networks*, and *Operations Research Letters*, and is a member of several professional societies including AMS, ACM, MAA, MPS, NYAS, ORSA, and SIAM.

**Diane D. Sheng** received the B.S. degree in math sciences, the M.S. degree in statistics, and the Ph.D. degree in operations research from Stanford University, Stanford, CA, in 1974, 1976, and 1978, respectively.

She joined AT&T Bell Laboratories, Holmdel, NJ, in December 1978 where she is a Supervisor, Department of Data Network Analysis. Her work has included research on stochastic models and control problems, the performance analysis of PBX systems, and the design, traffic engineering, and planning of various AT&T data networks.