[29] L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," in *AFIPS SJCC Proc.*, vol. 36, p. 543, May 1970.
[30] M. Shaw, W. A. Wulf and R. L. London, "Abstraction and verification in Alphard: Defining and specifying iteration and generators," *CACM*, vol. 20, no. 8, p. 553, Aug. 1977.
[31] R. F. Sproull, "Omnigraph-Simple terminal-independent graphics software," Xerox Palo Alto Res. Center, CSL-73-4, 1973.
[32] —, "InterLisp display primitives," Xerox Palo Alto Res. Center, 1977, informal note.
[33] R. F. Sproull and E. L. Thomas, "A network graphics protocol," *Comput. Graphics*, vol. 8, no. 3, Fall 1974.
[34] C. A. Sunshine, "Survey of protocol definition and verification techniques," in *Computer Network Protocols*, A. Danthine, Ed. Liege, Belgium, Feb. 1978.
[35] W. Teitelman, "A display oriented programmer's assistant," Xerox Palo Alto Res. Center, CSL-77-3, 1977.
[36] R. H. Thomas, "A resource sharing executive for the ARPAnet," in *AFIPS Proc.*, NCC, p. 155, 1973.
[37] —, "MSG: The interprocess communication facility for the national software works," Bolt Beranek and Newman, Rep. 3483.
[38] D. C. Walden, "A system for interprocess communication in a resource-sharing computer network," *CACM*, vol. 15, no. 4, p. 221, Apr. 1972.
[39] J. E. White, "A high-level framework for network-based resource sharing," *AFIPS Proc.*, NCC, p. 561, 1976.
[40] P. A. Woodsford, "The design and implementation of the GINO 3D graphics software package," *Software Practice and Experience*, vol. 1, p. 335, Oct. 1971.

Issues in Packet-Network Interconnection

VINTON G. CERF AND PETER T. KIRSTEIN

Invited Paper

Abstract—**This paper introduces the wide range of technical, legal, and political issues associated with the interconnection of packet-switched data communication networks. Motivations for interconnection are given, desired user services are described, and a range of technical choices for achieving interconnection are compared. Issues such as the level of interconnection, the role of gateways, naming and addressing, flow and congestion control, accounting and access control, and basic internet services are discussed in detail. The CCITT X.25/X.75 packet-network interface recommendations are evaluated in terms of their applicability to network interconnection. Alternatives such as datagram operation and general host gateways are compared with the virtual circuit methods. Some observations on the regulatory aspects of interconnection are offered and the paper concludes with a statement of open research problems and some tentative conclusions.**

I. Introduction

IT IS THE THEME of many papers in this issue, that people need access to data resources. In many cases this access must be over large distances, in others it may be local to a building or a single site. Data networks have been set up to meet many user needs—often, but not necessarily, using packet-switching technology. For single organizations, these data networks are often private ones, built with a technology optimized to the specific application. For communication between organizations, these networks are being set up by licensed carriers. In North America, there are many such licensed carriers, e.g., TELENET [1], DATAPAC [2], and TYMNET [3]. In the rest of the world, the Post, Telegraph, and Telephone Authority (PTT) in each country has a near monopoly on such services; special public data networks being set up in these countries include TRANSPAC [5] in France, EURONET [6] for inter-European traffic, DDX [7] in Japan, EDS [8] in the Federal Republic of Germany, and the Nordic Public Data Network (NPDN, [9]) in Scandinavia. These public data networks are considered in greater detail in other references (e.g., [10]-[12]). Most of the above networks use packet-switching technology; some of them, e.g., EDS and the NPDN, do not do so yet, but may do so in the future. In some cases special data networks have been authorized for specific communities, e.g., SITA [13] for the airlines, and SWIFT [14] for the banks. In addition many private networks have been set up among individual organizations, and experimental networks of different technologies have been developed also, e.g., ARPANET [15], [16], CYCLADES [17], ETHERNET [18], SPYDER [19], PRNET [20], [21] and SATNET [22].

Manuscript received June 20, 1978; revised July 21, 1978.
V. G. Cerf is with the Advanced Research Projects Agency, U.S. Department of Defense, Arlington, VA 22209.
P. T. Kirstein is with the Department of Statistic and Computer Science, University College, London, England.

It is a common user requirement that a single terminal and access port should be able to access any computing resource the user may desire—even if the resource is on another data network. From this requirement, there is a clear user need to have data networks connected together. By the same token, the providers of data network services would like to have their networks used as intensively as possible; thus they also have a strong motivation to connect their data networks to others. As a result of these considerations, there has been a high recent interest in the issues arising in the connection of data networks [23]-[26], [32].

From the user viewpoint, the requirement for interconnection of data networks is independent of the network technology. From the implementation viewpoint, there can be some considerable complications in connecting networks of widely different technologies—such as circuit-switched and datagram packet-switched networks (these terms are explained below). On the whole we will consider only, in this paper, the interconnection of packet-switched data networks. In many cases, however, the arguments will be equally valid for the interconnection of packet-switched to circuit-switched networks.

Network interconnection raises a great many technical, legal, and political questions and issues. The technical issues generally revolve around mechanisms for achieving interconnection and their performance. How can networks be interconnected so that packets can flow in a controllable way from one net to another? Should all computer systems on all nets be able to communicate with each other? How can this be achieved? What kind of performance can be achieved with a set of interconnected networks of widely varying internal design and operating characteristics? How are terminals to be given access to resources in other networks? What protocols are required to achieve this? Should the protocols of one net be translated into those of another, or should common protocols be defined? What kinds of communication protocol standards are needed to support efficient and useful interconnection? Who should take responsibility for setting standards?

The legal and political issues are at least as complex as the technical ones. Can private networks interconnect to each other or must they do so through the mediation of a public network? How is privacy to be protected? Should there be control over the kinds of data which move from one net to another? Are there international agreements and conventions which might be affected by international interconnection of data networks? What kinds of charging and accounting policies should apply to multinetwork traffic? How can faults and errors be diagnosed in a multinet environment? Who should be responsible for correcting such faults? Who should be responsible for maintaining the gateways which connect nets together?

We cannot possibly answer all of these questions in this paper, but we deal with many of them in the sections below.

This paper is divided into eleven sections. In the next section we provide some definitions, and in Section III we explore some of the motivations for network interconnection. In Section IV we discuss the range of end-user service requirements and choices for providing multinetwork service. Section V reviews the concept of computer-communication protocol layering. Section VI reviews the basic interconnection choices and introduces the concept of gateways between nets, protocol translation and the impact of common protocols; it elaborates also on the function of gateways. Section VII discusses the CCITT recommendations X.25 and X.75 and their role in network interconnection. Section VIII describes some of the network interconnections achieved and some of the experiments in progress. Section IX outlines regulatory issues raised by network interconnection alternatives. Section X mentions some unresolved research questions, and the final section offers some tentative conclusions on network interconnection issues.

II. The Definition of Terms

The vocabulary of networking is extensive and not always consistent. We introduce some generic terms below which we will use in this paper for purposes of discussion. It is important for the reader not to make any *a priori* assumptions about the physical realization of the objects named or of the boundary of jurisdictions owning or managing them. For instance, a gateway (see below) might be implemented to share the hardware of a packet switch and be owned by a packet-switching service carrier; alternatively it might be embedded in a host computer which subscribes to service on two or more computer networks. Roughly speaking, we are assigning names to groups of functions which may or may not be realized as physically distinct entities.

Packet: A packet of information is a finite sequence of bits, divided into a control header part and a data part. The header will contain enough information for the packet to be routed to its destination. There will usually be some checks on each such packet, so that any switch through which the packet passes may exercise error control. Packets are generally associated with internal packet-network operation and are not necessarily visible to host computers attached to the network.

Datagram: A finite length packet of data together with destination host address information (and, usually, source address) which can be exchanged in its entirety between hosts, independent of all other datagrams sent through a packet switched network. Typically, the maximum length of a datagram lies between 1000 and 8000 bits.

Gateway: The collection of hardware and software required to effect the interconnection of two or more data networks, enabling the passage of user data from one to another.

Host: The collection of hardware and software which utilizes the basic packet-switching service to support end-to-end interprocess communication and user services.

Packet Switch: The collection of hardware and software resources which implements all intranetwork procedures such as routing, resource allocation, and error control and provides access to network packet-switching services through a host/network interface.

Protocol: A set of communication conventions, including formats and procedures which allow two or more end points to communicate. The end points may be packet switches, hosts, terminals, people, file systems, etc.

Protocol Translator: A collection of software, and possibly hardware, required to convert the high level protocols used in one network to those used in another.

Terminal: A collection of hardware and possibly software which may be as simple as a character-mode teletype or as complex as a full scale computer system. As terminals increase in capability, the distinction between "host" and "terminal" may become a matter of nomenclature without technical substance.

Virtual Circuit: A logical channel between source and destination packet switches in a packet-switched network. A

virtual circuit requires some form of "setup" which may or may not be visible to the subscriber. Packets sent on a virtual circuit are delivered in the order sent, but with varying delay.

PTT: Technically PTT stands for Post, Telegraph, and Telephone Authority; this authority has a different form in different countries. In this paper, by PTT we mean merely the authority (or authorities) licensed in each country to offer public data transmission services.

We have attempted to make these definitions as noncontroversial as possible. For example, in the definition of packet switch, we alluded to a host/network interface. The reader should not assume that subscriber services are limited to those offered through the host/network interface. The packet-switching carrier might also offer host-based services and terminal access mechanisms as additional subscriber services.

III. The Motivating Forces in the Interconnection of Data Networks

In the introduction, we mentioned that there was a strong interest, among both the users and suppliers of data serivces, in the interconnection of data networks. However, the technical interests of the different parties are not identical. The end user would merely like to be able to access any resources from a single terminal, with a single access port, as economically as possible according to his own performance criteria. A Public Carrier, or PTT, has a strong motivation to connect its network to other PTT's. As in the telephone system, the concept of all subscribers being accessible through a single Public Data Service, is considered highly desirable; however the different PTT's may have restricted geographic coverage, or only a specific market penetration.

The motivation of the PTT's to interface to private networks is weaker and more complex. They always provide facilities to attach single terminals, where a terminal may be a complex computer system; they are often not interested, at present, in making any special arrangements when the "terminal" is a whole computer network. The operators of private networks often have a vital interest in connecting their networks to other private networks and to the public ones. Even though in many cases the bulk of its traffic is internal to the private network, which is why it was set up in the first place, there is usually a vital need to access resources not available on that network. The regulatory limitations often imposed on the method of interconnection of private networks are discussed in Section IX. In some countries, it is not permitted to build private networks using leased line services, but intrabuilding networks may be permitted. Interconnection of such local networks to public networks may play a crucial role in making the local network useful.

To date the PTT's have tried to standardize on access procedures for their Public-Packet Data Services. The standardization has taken place in the International Consultative Committee on Telegraphy and Telephony (called CCITT) in a set of recommendations called X.3, X.25, X.28, and X.29 ([27]-[29]). Not all PTT's have such forms of access yet, but most of the industrialized nations in the West are moving in this direction. This series of recommendations is discussed in much more detail in Section VI; it does not pay special attention to the attachment of private networks ([31], [32]), but the recommendations are themselves expected to change to meet this requirement. The PTT's are agreeing on a set of interface recommendations and procedures called X.75 [33], to connect their networks to each other; so far this interface procedure (and its corresponding hardware) is not intended to be provided to private networks.

While most PTT's have preferred to ignore the technical implications of the attachment of private networks to the public ones, most private network operators cannot ignore this requirement. They are often motivated to add some extra "Foreign Exchange" capability as an afterthought, with minimum change to their intranetwork procedures; this approach can be successful up to a point, but will usually be limited by the lack of high-level procedures between the different networks. These high-level procedures have not yet been considered by CCITT, but it has been proposed that CCITT Study Group VII investigate high-level procedures and architectural models, in cooperation with the investigation of "open system architectures" by Technical Committee 97, Sub-Committee 16 of the International Standards Organisation (ISO). This subject is also considered later in this paper, in Section VI.

An aim of these standardization exercises is to ensure that both manufacturer and user implementations of network resources can communicate with each other through single private or public data networks. A consequence should be that the resources are also compatibly accessible over connected data networks.

Depending on the applications and spatial distribution of subscribers, the preferred choice of packet-switching medium will vary. Intrabuilding applications such as electronic office services may be most economically provided through the use of a coaxial-packet cable system such as the Xerox ETHERNET [18] and LCSNET [64], or twisted pair rings such as DCS [34], coupled with a mix of self-contained user computers (e.g., intelligent terminals with substantial computing and memory capacity) and shared computing, storage, and input-output facilities. Larger area regional applications might best employ shared video cables [35] or packet radios [20], [21] for mobile use. National systems might be composed of a mixture of domestic satellite channels and conventional leased-line services. International systems might use point-to-point links plus a shared communication satellite channel and multiple ground stations to achieve the most cost-effective service.

A consequence of the wide range of technologies which are optimum for different packet-switching applications is that many different networks, both private and public, may co-exist. A network interconnection strategy, if properly designed, will permit local networks to be optimized without sacrificing the possibility of providing effective internetwork services. The potential economic and functional advantages of local networks such as ETHERNET or DCS will lead naturally to private user networks. Such private network developments are analogous to telephone network private automated branch exchanges (PABX) and represent a natural consequence of the marriage of computer and telecommunication technology.

Two further developments can be expected. First, organizations which are dispersed geographically, nationally, or internationally, will want to interconnect these private networks both to share centralized resources and to effect intraorganization electronic mail and other automated office services. Second, there will be an increasing interest in interorganization interconnections to allow automated procurement and financial transaction services, for example, to be applied to interorganization affairs.

In most countries where private networks are permitted, interorganization telecommunication requires the involvement of a PTT. Hence the most typical network interconnection

scenarios will involve three or four networks. Within one national administration the private nets of different organizations will be interconnected through a public network. International interconnections will involve at least two public networks. We will return to this topic in Section VI.

In addition to permitting locally optimized networks to be interconnected, a network interconnection strategy should also support the gradual introduction of new networking technology into existing systems without requiring simultaneous global change throughout. This consideration leads to the conclusion that the public data networks should support the most important user requirements for internet service from the outset. If this were the case, then changes in network technology which require a multinetwork system during phased transition would not, *a priori*, have to affect user services.

IV. Provision of End-User Multinetwork Services

The ultimate choice of a network interconnection strategy will be strongly affected by the types of user services which must be supported. It is useful to consider the range of existing and foreseeable user service requirements without regard for the precise means by which these requirements are to be met. We will leave for discussion in subsequent sections the choice of supporting the various services within or external to the packet-switched network. The types of service discussed below are general requirements for network facilities. For this reason they also should be supported across interconnected networks.

Most of the currently prevalent computer-communication services fall into four categories:

1) terminal access to time-shared host computers;
2) remote job entry services (RJE);
3) bulk data transfer;
4) transaction processing.

The time-sharing and transaction services typically demand short network and host response times but modest bandwidth. The RJE and file transfer services more often require high amounts of data transfer, but can tolerate longer delay. Some networks were designed to support primarily terminal service, leaving RJE or file transfer services to be supported by dedicated leased lines. Packet-switching techniques permit both types of service to be supported with common network resources, leading to verifiable economies. However, bulk data transfer requires increasingly higher throughput rates if delivery delays are to be kept constant as the amount of data to be transferred increases.

As distributed operating systems become more prevalent, there will be an increased need for host-to-host transaction services. A prototypical example of such a system is found in the DARPA National Software Works [4], [36]. In such a system, small quantities of control information must be exchanged quickly to coordinate the activity of the distributed components. Broadcast or multidestination services will be needed to support distributed file systems in which information can be stored redundantly to improve the reliability of access and to protect against catastrophic failures.

Transaction services are also finding application in reservation systems, credit verification, point of sale, and electronic funds-transfer systems in which hundreds or thousands of terminals supply to, or request of, hosts small amounts of information at random intervals. Real-time data collection for

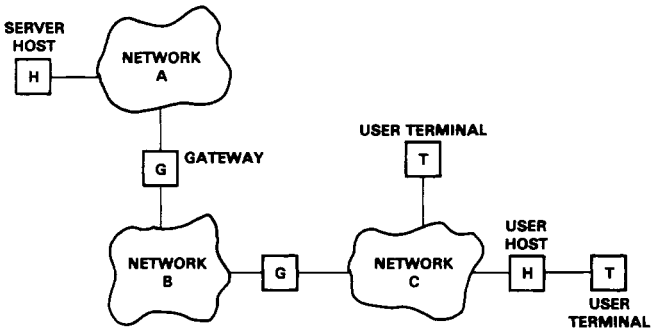

Fig. 1. Network concatenation.

weather analysis, ground and air traffic control, and meter reading, for example, also fall into this category.

More elaborate user requirements can be foreseen as electronic mail facilities propagate. Multiple destination addressing and end-to-end encryption for the protection of privacy as well as support for text, digitized voice, and facsimile message transmission are all likely requirements. Electronic teleconferencing using mixtures of compressed digital packet speech, videographics, real-time cursors (for pointing at video images under discussion), and text display will give rise to requirements for closed user groups and time-synchronized mixes of transaction-like (e.g., for cursor tracking and packet speech) and reliable circuit-like services (e.g., for display management).

Reliability and rapid response will be increasingly important as more and more computer-based applications requiring telecommunications are integrated into the business, government, military, and social fabric of the world economy. The more such systems are incorporated into their daily activities, the more vulnerable the subscribers are to failures. Reliability concerns lead to the requirement for redundant alternatives such as distributed file systems, richly connected networks, and substantial local processing and storage capability. These trends increase the need for networking to share common hardware and software resources (and thus reduce their marginal cost), to support remote software maintenance and debugging, and to support intra- and inter-organizational information exchange.

We have described the end-user services required across one or more data networks. We have carefully refrained from discussing which services should be provided in the data network, and which should be provided in the hosts. Here the choice in single networks will depend on the network technology and the application requirements. For example, in a network using a broadcast technology such as ETHERNET or the SATNET, multidestination facilities may well be incorporated in the data network itself. In typical store-and-forward networks, this feature might be provided at the host level by the transmission of multiple copies of packets. This example highlights immediately the difficulty of using sophisticated services at the data network level across concatenated networks. If A, B, and C are data networks connected as in Fig. 1, and A and C but not B support broadcast or real-time features, it is very difficult to provide them across the concatenation of A, B, and C.

The problem of achieving a useful set of internetwork services might be approached in several ways, as follows.

1) Require all networks to implement the entire range of desired services (e.g., datagram, virtual circuit, broadcast, real-

time, etc.), and then attempt to support these services across the gateways between the networks.

2) Require all networks to implement only the most basic services (e.g., datagram or virtual circuit), support these services across gateways, and rely on the subscriber to implement all other services end-to-end.

3) Allow the subscriber to identify the services which he desires and provide error indications if the networks involved, or the gateways between them, cannot provide the desired services.

4) Allow the subscriber to specify the internetwork route to be followed and depend on the subscriber to decide which concatenation of services are appropriate and what end-to-end protocols are needed to achieve the ultimately preferred class of service.

5) Provide one set of services for local use within each network and another, possibly different set for internetwork use.

The five choices above are by no means exhaustive, and, in fact, only scratch the surface of possibilities. Nothing has been said, thus far, about the compatibility of various levels of communication protocols which exist within each network, within subscriber equipments, and within the logical gateway between networks. To explore these issues further, it will be helpful to have a model of internetwork architecture, taking into account the common principle of protocol layering and the various possible choices of interconnection strategy which depend upon the protocol layer at which the networks are interfaced. We consider this in the next section.

V. Layered Protocol Concepts

Both to provide services in single networks, and to compare the capabilities of different networks, a very useful concept in networking is protocol layering. Various services of increasing capability can be built one on top of the other, each using the facilities of the service layer below and supporting the facilities of the layer above. A thorough tutorial on this concept can be found in the paper by Pouzin and Zimmermann in this issue [37]. We give some specific examples below of layering as a means of illustrating the scope of services and interfaces to be found in packet networks today—and some of the problems encountered in offering services across multiple networks.

Table I offers a very generic view of a typical protocol hierarchy in a store-and-forward computer network, including layers usually found outside of the communication network itself. There are several complications to the use of generic protocol layering to study network interconnection issues. Chief among these is that networks do not all contain the same elements of the generic hierarchy. A second complication is that some networks implement service functions at different protocol layers. For instance, virtual circuit networks implement an end/end subscriber virtual circuit in their intranet, end/end level protocol. Finally, the hierarchical ordering of functions is not always the same in all networks. For instance, TYMNET places a terminal handling protocol within the network access layer, so that hosts look to each other like one or more terminals. Figs. 2–7 illustrate the functional layering of some different networks. It is important to note how the functions vary with the choice of transmission medium.

TABLE I
Generic Protocol Layers

PROTOCOL LAYER	FUNCTIONS
6. APPLICATION	FUNDS TRANSFER, INFORMATION RETRIEVAL, ELECTRONIC MAIL, TEXT EDITING . . .
5. UTILITY	FILE TRANSFER, VIRTUAL TERMINAL SUPPORT
4. END/END SUBSCRIBER	INTERPROCESS COMMUNICATION (E.G. VIRTUAL CIRCUIT, DATAGRAM, REAL-TIME, BROADCAST)
3. NETWORK ACCESS	NETWORK ACCESS SERVICES (E.G. VIRTUAL CIRCUIT, DATAGRAM . . .)
2. INTRANET, END-TO-END	FLOW CONTROL, SEQUENCING
1. INTRANET, NODE-TO-NODE	CONGESTION CONTROL, ROUTING
0. LINK CONTROL	ERROR HANDLING, LINK FLOW CONTROL

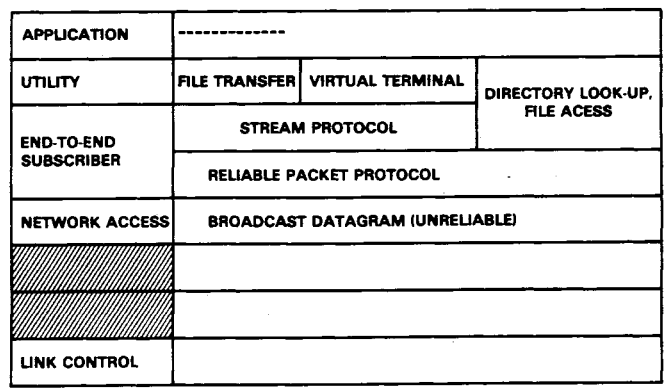

Fig. 2. ETHERNET protocol layering.

A. ETHERNET

In Fig. 2, we represent the Xerox ETHERNET protocol hierarchy. The basic link control mechanism is the ability of the interface device to detect conflict on a shared coaxial cable. If a transmitting interface detects that another interface is also transmitting, it immediately aborts the transmission. Hosts attached to the network interface present datagrams to be transmitted and are told if the datagram was aborted. Datagrams can be addressed to specific interfaces or to all of them. The end/end subscriber layer of protocol is split into two parts: a reliable datagram protocol in which each datagram is reliably delivered and separately acknowledged, and a stream protocol which can be thought of as a virtual circuit. This split is possible, in part, because there is a fairly large maximum datagram size (about 500 bytes) so that user applications can send datagrams without having to fragment and reassemble them. This makes the datagram service useful for many applications which might otherwise have to use the stream protocol. All higher level protocols, such as Virtual Terminal and File Transfer, are carried out in the hosts.

B. ARPANET

The ARPANET protocol hierarchy is shown in Fig. 3. The basic link control between packet switches treats the physical link as eight independent virtual links. This increases effective throughput, but does not necessarily preserve the order in which packets were originally introduced into the network. The intranet node-to-node protocols deal with adaptive routing decisions, store-and-forward service, and congestion control. Hosts have the option of either passing messages (up to

APPLICATION	RJE	ELECTRONIC MAIL	
UTILITY	TELNET	FTP	
END/END SUBSCRIBER	NCP	TCP	NVP/NVCP
NETWORK ACCESS	PERMANENT VIRTUAL CIRCUIT		DATAGRAM
INTRANET, END/END	FLOW CONTROL, SEQUENCING, MESSAGE REASSEMBLY		
INTRANET, NODE/NODE	ADAPTIVE ROUTING, STORE AND FORWARD, CONGESTION CONTROL		
LINK CONTROL	NON-SEQUENCED, MULTI-CHANNEL ERROR CONTROL		

Fig. 3. ARPANET protocol layering.

END/END SUBSCRIBER	TERMINAL-TO-HOST
NETWORK ACCESS	VIRTUAL CIRCUIT
INTRANET END-END	
INTRANET NODE-NODE	FRAME DISASSEMBLY, REASSEMBLY, ROUTING, STORE/FORWARD, CONGESTION CONTROL
LINK CONTROL	FRAME-BASED ERROR CONTROL, RETRANSMISSION, SEQUENCING

Fig. 4. TYMNET protocol layering.

8063 bits of text) across the host/network interface, which will be delivered in sequence to the destination, or passing datagrams (up to 1008 bits of text) which are not necessarily delivered in sequence. The user's network access interface is datagram-like in the sense that no circuit setup exchange is needed even to activate the sequenced message service. In effect, this service acts like a permanent virtual circuit over which a sequence of discrete messages are sent. For the sequenced messages, there is exactly one virtual circuit maintained for each host/host pair. In fact, these virtual circuits are set up dynamically and terminated by the source/destination packet switches so as to improve resource utilization [38], [62].

The end/end subscriber layer of ARPANET contains two main protocols: Network Control Protocol (NCP, [39], [40]) and Transmission Control Protocol (TCP, [25]). NCP was the first interprocess communication protocol built for ARPANET. It relies on the sequenced message service provided by the network and derives multiple virtual circuits between pairs of hosts by multiplexing. The TCP can use either the sequenced message service or the datagram service. It does its own sequencing and end/end error control and derives multiple virtual circuits through extended addressing and multiplexing. TCP was designed for operation in a multinet environment in which the only service which reasonably could be expected was an unreliable, unsequenced datagram service.

To support experiments in packetized voice communication, two protocols were developed for use on the ARPANET. The Network Voice Protocol (NVP) and Network Voice Conferencing Protocol (NVCP) use the datagram service to achieve very low delay and interarrival time variance in support of digital, compressed packet speech (more on these protocols may be found in [41]). The NVP could be considered the basis for a generic protocol which could support a variety of real-time, end/end user applications.

The higher level utility protocols such as terminal/host protocol (TELNET, [40], [42]) and file transfer protocol (FTP, [40], [42]) use virtual circuits provided by NCP or TCP. The FTP requires one live interactive stream to control the data transfer, and a second for the data stream itself. Yet higher level applications such as electronic mail and remote job entry (RJE, [40], [42]) use mixtures of TELNET and FTP to effect the service desired. These protocols are usually put into the hosts. There is one anomaly, which occurs in many networks. Because terminal handling is required so frequently, a Terminal Interface Message Processor (TIP, [43]) was built. This device is physically integrated with the packet switch (IMP, [38]); it includes also the NCP and TELNET protocols.

C. TYMNET

TYMNET (see Fig. 4) is one of the oldest of the networks in the collection described here [3]. Strictly speaking, it operates rather differently than other packet-switched networks, because the frames of data that move from switch to switch are disassembled and reassembled in each switch as an integral part of the store-and-forward operation. Nevertheless, the network benefits from the asynchronous sharing of the circuits between the switches in much the same way that more typical packet-switched networks do. The network was designed to support remote terminal access to time-shared computer resources. The basic service is the transmission of a stream of characters between the terminal and the serving host. A frame is made up of one or more blocks of characters, each block labeled with its source terminal identifier and length. The switch-to-switch layer of protocol disassembles each frame into its constituent blocks and uses a routing table to determine to which next switch the block should be sent. Blocks destined for the same next switch are batched together in a frame which is checksummed and sent via the link control procedure to the next switch. Batching the blocks reduces line overhead (the blocks share the frame checksum) at the expense of more CPU cycles in the switch for frame disassembly and reassembly.

The protocol between TYMNET switches also includes a flow control mechanism which, because of the fixed routes, can be used to apply back pressure all the way back to the traffic source. This is not precisely an end-to-end flow control mechanism, but a hop-by-hop back pressure strategy. Character blocks are kept in sequence along the fixed routes so that no resequencing is required as they exit from the network at their destinations. The network interface is basically a virtual circuit designed to transport character streams between a host and a terminal. The same virtual circuits can be used to transport character streams between hosts, which look to each other like a collection of terminals. Above the basic virtual circuit service, is a special echo-handling protocol which allows the host and the terminal handler in the "remote TYMSAT" to coordinate the echoing of the characters typed by a user.

D. PTT Networks

Many PTT networks, e.g., TELNET, TRANSPAC, DATAPAC, and EURONET use a particular network-access protocol, X.25 [28], [29] (see Fig. 5). This protocol has been recommended by the CCITT for public packet-switched data networks. X.25 is a three-part protocol consisting of a hardware electrical interface, X.21 [44], the digital equivalent of the usual V.24 or EIA-RS232C modem interface [45], a link control procedure, High Level Data Link Control (HDLC, [46]), and a packet-level protocol for effecting the setup, use, termination, flow, and error control of virtual circuits.

UTILITY	TERMINAL HANDLING X.28, X.29
END/END SUBSCRIBER	
NETWORK ACCESS	X.25, PERMANENT OR TEMPORARY VIRTUAL CIRCUITS
INTRANET, END-END	MULTIPLE VIRTUAL CIRCUITS, FLOW CONTROL
INTRANET NODE-NODE	ROUTING, STORE/FORWARD, CONGESTION CONTROL
LINK CONTROL	HDLC OR EQUIVALENT

Fig. 5. PTT protocol layering.

In all but the DATAPAC network, a fixed route for routing packets through the network is selected at the time the virtual circuit is created. "Permanent" virtual circuits are a customer option; if used, the setup phase is invoked only in the case of a network failure. Between source and destination packet switches, a virtual circuit protocol is operated which implements end-to-end flow control on multiple virtual circuits between pairs of packet switches. Up to 4096 virtual circuits between pairs of host ports can be maintained by each packet switch, as compared to the single virtual circuit provided by ARPANET (on which hosts can multiplex their own virtual circuits). This choice has a noticeable impact on the subscriber interface protocol which becomes complicated because the subscriber host and the packet switch to which it attaches must maintain a consistent view of the state of each virtual circuit in use.

To provide for echo control, user commands, code conversion, and other terminal-related services, these networks implement CCITT Recommendations X.28 [29] and X.29 [29] in a PAD (Packet Assembly and Disassembly unit). These protocols sit atop the virtual circuit X.25 protocol. In order to serve customers desiring a terminal-to-host service with character terminals, such as is provided by TYMNET or by the ARPANET (through the TIP), most of the PTT networks mentioned are developing a PAD unit. A matching X.29 (PAD control protocol) layer must be provided in hosts offering to service terminals connected to PAD's.

E. High Level Protocols

The X.25/X.28/X.29 protocol hierarchy does not include an end/end subscriber or high-level protocol layer. Some customers will, in fact, implement end-to-end protocols on top of the virtual circuit protocol, but others may not. Several attempts are being made to standardize protocols above the network access level. The ARPANET community has developed a Transmission Control Protocol [25] for internetwork operation to replace the Network Control Program (NCP) developed early in the ARPANET project. The International Federation of Information Processing (IFIP) has proposed a Transport Station through its Working Group 6.1 on Network Interconnection [47]; the proposal has been submitted to the International Standards Organisation (ISO) as a draft standard. In addition, other communities, e.g., the High Level Protocol Working Group in the UK, have devised protocols for Virtual Packet Terminals (VPT, [48]) and File Transport Protocol (FTP, [49]) which are intended to be network independent and which may be submitted to CCITT. The ISO study on "open systems architecture" and the proposed similar study by CCITT Study Group VII will attempt to evolve higher level protocol recommendations for existing and future data networks.

This brief summary of different network-protocol layerings is in no way comprehensive, but illustrates the diversity of protocol designs which can be found on nets providing different types of services to subscribers.

VI. Technical Interconnection Choices

A. The Issues

Beginning with the earliest papers dealing with strategies for packet-network interconnection [23]-[26], [32], the common objective of all the proposed methods is to provide the physical means to access the services of a host on one network to all subscribers (including hosts) of all the interconnected networks. Of course, limitations to this accessibility are envisaged, imposed either for administrative reasons or by the scarcity of resources. The achievement of this objective invariably requires that data produced at a source in one net be delivered and correctly interpreted at the destination(s) in another network. In an abstract sense, this boils down to providing interprocess communication across network boundaries. Even if a person is the ultimate source of the data, packet-switching networks must interpose some degree of software processing between the person and the destination service, even if only to assemble or disassemble packets produced by a computer terminal.

A fundamental aspect of interprocess communication is that no communication can take place without some agreed conventions. The communicating processes must share some physical transmission medium (wire, shared memory, radio spectrum, etc.), and they must use common conventions or agreed upon translation methods in order to successfully exchange and interpret the data they wish to communicate. One of the key elements in any network interconnection strategy is therefore how the required commonality is to be obtained. In some cases, it is enough to translate one protocol into another. In others, protocols can be held in common among the communicating parties.

In any real network interconnection, of course, a number of secondary objectives will affect the choice of interconnection strategy. For example achievable bandwidth, reliability, robustness (i.e., resistance to failures), security, flexibility, accountability, access control, resource allocation options, and the like can separately and jointly influence the choice of interconnection strategy. Combinations of strategies employing protocol standards and protocol translations at various levels of the layered protocol hierarchy are also likely possibilities.

There are a number of issues which must be resolved before a coherent network interconnection strategy can be defined. A list of some of these issues, which will be treated in more detail in succeeding sections, is:

1) level of interconnection;
2) naming, addressing, and routing;
3) flow and congestion control;
4) accounting;
5) access control;
6) internet services.

B. Gateways and Levels of Network Interconnection

The concept of a gateway is common to all network interconnection strategies. The fundamental role of the gateway is to terminate the internal protocols of each network to which it is attached while, at the same time, providing a common

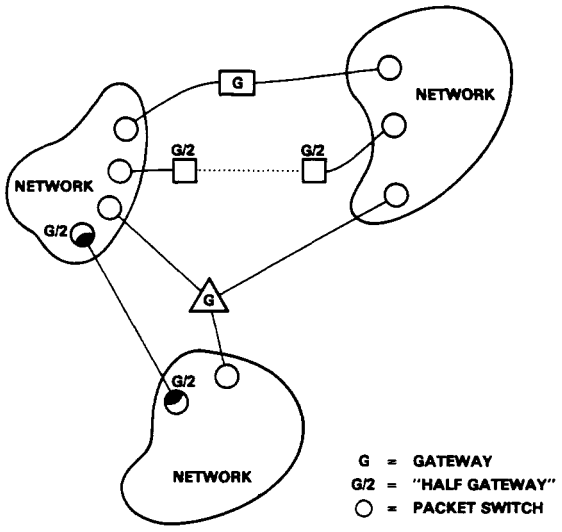

Fig. 6. Various gateway configurations.

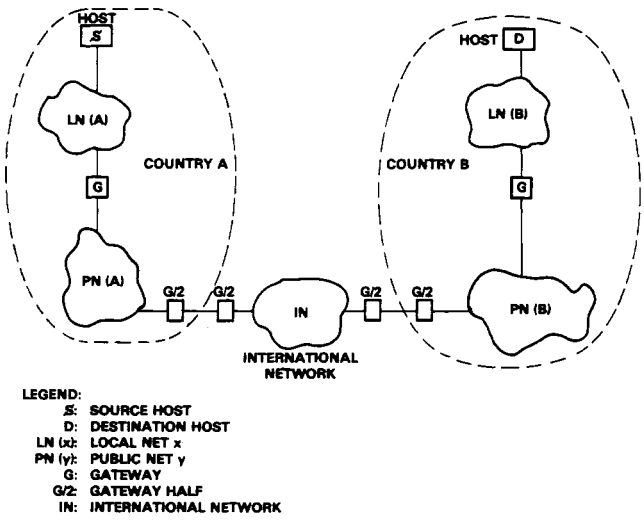

Fig. 7. International packet-networking model.

ground across which data from one network can pass into another. However, the choice of functions to be performed in the gateway varies considerably among different interconnection strategies (see Fig. 6). The term "gateway" need not imply a monolithic device which joins a pair of networks. Indeed, the gateway may merely be software in a pair of packet switches in different networks, or it may be made up of two parts, one in each network (a sort of "gateway half"). In the latter case, the two halves might be devices separate and distinct from the network packet switches or might be integrated with them. Furthermore, a gateway might interconnect more than two networks. In the material which follows, every attempt has been made to avoid any implicit choice of gateway implementation. It is worth pointing out, however, that the "half gateway" concept is highly attractive from both a technical and a purely administrative point of view. Technically, each half could terminate certain levels of protocol of the net to which it is attached. Administratively each half could be the responsibility of the network to which it belongs. Then the only matters for jurisdictional negotiation are the physical medium by which the half-gateways exchange data, and the format and protocol of the exchange.

It is important to realize that typical applications may involve three or more networks. Where local networks are used, they will usually need to be interconnected to realize the benefits of interorganizational data exchange. In most countries, such interconnections will only be permitted through a public network. Thus for a typical national situation, three networks and two gateways will be involved in providing the desired host-to-host communication.

The international picture is similar, except that more networks are likely to be involved. Shown in Fig. 7, the path from a host, S, on local network $LN(A)$ in country A, passes through a public network, $PN(A)$ in country A, through an international network IN, through a public network $PN(B)$ in country B, and finally through a local network, $LN(B)$, to the destination host, D. There are four internetwork gateways involved. It is this model involving multiple gateways that guides us away from network interconnection methods which rely on the source and destination hosts being in adjacent networks connected by the mediation of a single gateway.

1) Common Subnet Technology (Packet Level Interconnection): The level at which networks are interconnected can be determined by the protocol layers terminated by the gateway. For example, if a pair of identical networks were to be interconnected at the interpacket-switch level of protocol, we might illustrate the gateway placement as shown in Fig. 8. Here the "gateway" may consist only of software routines in the adjacent packet switches, e.g., $P(A)$ and $P(B)$, which provide accounting, and possibly readdressing functions. The contour model of protocol layer is useful here since it shows which levels are common to the two networks and which levels could be different. In essence, those layers which are terminated by the gateways could be different in each net, while those which are passed transparently through the gateway are assumed to be common in both networks. This network interconnection strategy requires that the internal address structure of all the interconnected networks be common. If, for example, addresses were composed of a network identifier, concatenated with a packet-switch identifier and a host identifier, then addressing of objects in each of the networks would be straightforward and routing could be performed on a regional basis with the network identifiers acting as the regional identifiers, if desired. Alternatively, two identical networks could adopt a common network name and assign nonduplicative addresses to each of the packet switches in both networks. This may require that addresses in one network be changed.

The strategy described above might be called the "common subnetwork strategy," since, in the end, subscribers of the newly formed joint network would essentially see a single network. This strategy does not rule out the provision of special access control mechanisms in the gateway nodes which could filter traffic flowing from one network into the other. Similarly, the gateway nodes could perform special internetwork traffic accounting which might not normally be performed in a subnet switching node. This network interconnection method is limited to those cases in which the nets to be connected are virtually identical, since the gateways must participate directly in all the subnet protocols. The end-to-end subnet protocols (e.g. source/destination packet-switch protocols) must pass transparently through the gateways to permit interactions between a source packet switch in one net and a destination packet switch in another. The resulting network presents the same network access interface to all

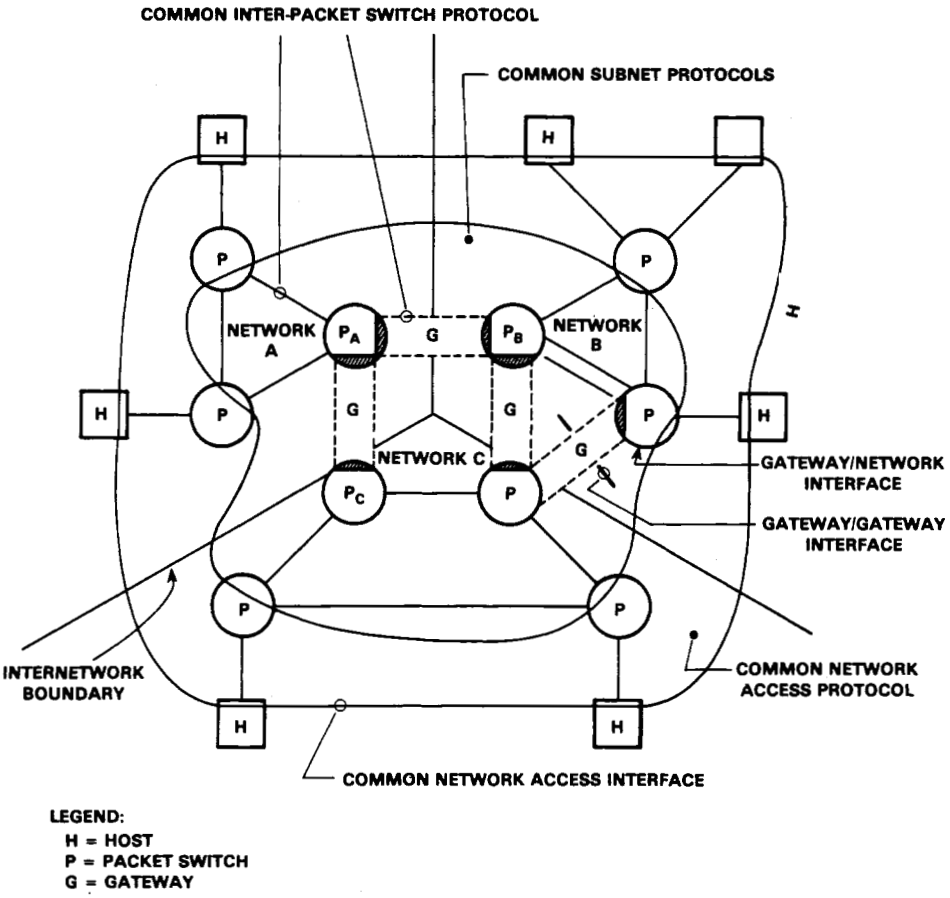

Fig. 8. Interconnection of common subnetworks.

subscribers, and this leads us to the next example which is based on the concept of a common network access interface.

2) Common Network Access Interfaces: If the subnetwork protocols are not identical, the next opportunity to establish internetwork commonality is at the network access interface. This is illustrated in Fig. 9. Each network is assumed to have its own intranet protocols. However, each network presents the same external interface to subscribers. This is illustrated by showing a common interface passing through all hosts, marked "common network access interface" in the figure.

Once again, the gateway could be thought of as software in adjacent packet switches. Each gateway is composed of two halves formed by linking the packet switches of two nets together. However, in this case, the subnetwork protocols are terminated at the gateway so that the intergateway exchange looks more like network access interaction than a node-to-node exchange. This is the approach taken by CCITT with its X.25 packet network interface recommendation and X.75 intergateway exchange recommendation.

It is important to note that the intergateway interface could be similar to the standard network access interface, but it need not necessarily be identical.

There are two basic types of network interface currently in use: 1) the datagram interface [31]; and 2) the virtual circuit interface [32]. The details of these generic interface types vary in different networks; some networks even offer both types of interface. In some, the interface to use may be chosen at subscription time; in others it may be possible for a subscriber to select the access method dynamically.

A datagram interface allows the subscriber to enter packets into the network independent of any other packets which have been or will be entered. Each packet is handled separately by the network. A virtual circuit interface requires an exchange of control information between the subscriber and the network for the purpose, for example, of setting up address translation tables, setting up routes or preallocating resources, before any data packets are carried to the destination. Some networks may implement a *fast select* virtual circuit interface in which a circuit setup request is sent together with the first (and possibly last) data packet. Other control exchanges would be used to close the resulting virtual circuits set up in this fashion.

It is essential to distinguish datagram and virtual circuit services from datagram and virtual circuit interfaces. A datagram service is one in which each packet is accepted and treated by the network independently of all others. Sequenced delivery is not guaranteed. Indeed, it may not be guaranteed that all datagrams will be delivered. Packets may be routed independently over alternate network paths. Duplicate copies of datagrams might be delivered.

Virtual circuit service tries to guarantee the sequenced delivery of the packets associated with the same virtual circuit. It typically provides to the host advice from the network on flow control per virtual circuit as opposed to the packet-by-packet acceptance or rejection typical of a datagram service. If the network operation might produce duplicate packets, these are filtered by the destination packet switch before delivery to the subscriber. Duplicate packet creation is a

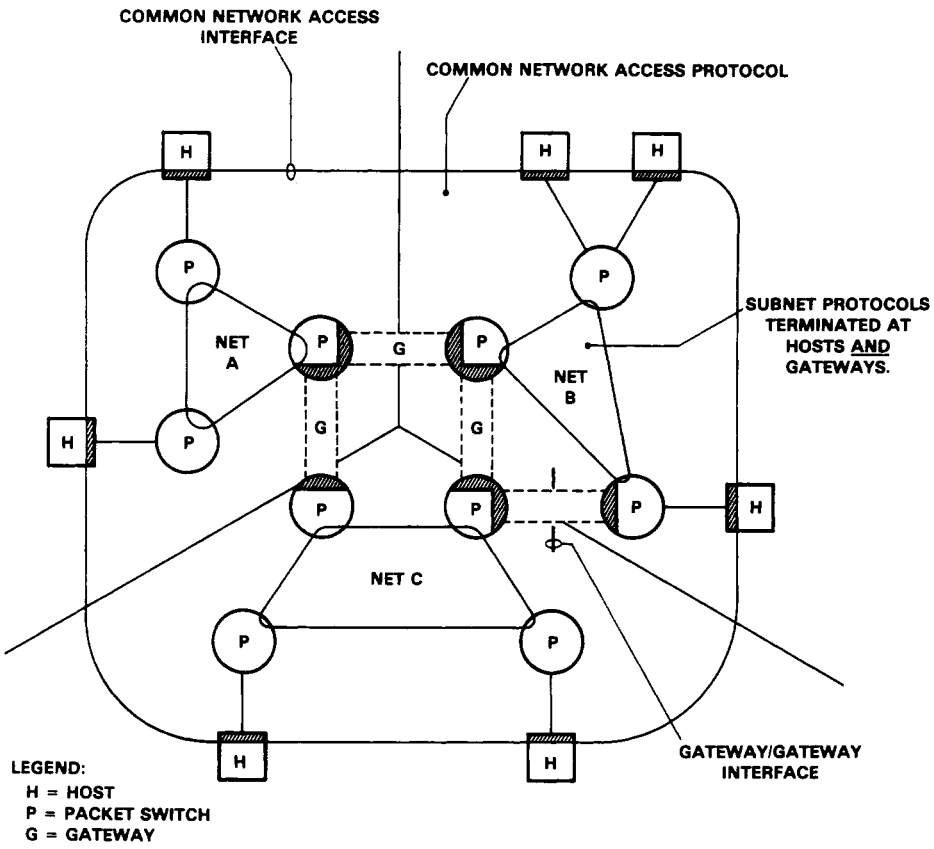

Fig. 9. Interconnection of networks with common network-access interfaces.

common phenomenon as in packet-switched store-and-forward systems. The basic mode of operation is to forward a packet to the next switch and await an acknowledgment. After a timeout, the packet is retransmitted. If an acknowledgment is lost due to line noise, for example, then two copies of the packet would have been transmitted. Even if the next switch is prepared to filter duplicates out, a network which uses adaptive routing can deliver a duplicate packet to the periphery of the network. For example, if a packet switch receives a packet successfully but the line to the sender breaks before the receiver can acknowledge, the sender may send another copy to a *different* packet switch. Both packet copies may be routed and delivered to the destination packet switch where final duplicate filtering would be needed if virtual circuit service is being provided.

Some networks offer both a datagram and a virtual circuit service; some offer a single interface, but different services. For example, the ARPANET has a basic datagram interface. However, the subnetwork will automatically provide a sequenced virtual circuit service (i.e., packets are kept in sequence when they are delivered to the destination) if the packet is marked appropriately. Otherwise, packets are not delivered in sequence nor are packet duplicates or losses, except for line by link correction, recovered within the network for nonsequenced types of traffic.

By contrast, TRANSPAC offers a virtual circuit interface and service. Subscribers transmit "call request" packets containing the full destination address to the packet switch. The request packet is forwarded to the destination, leaving behind a fixed route. The destination subscriber returns a "call accepted" packet which is delivered to the caller. As a result of this exchange, the source subscriber has associated a "logical channel number" or LCN, with the full source-destination addresses. Thus subsequent packets to be sent on the same logical channel are identified by the LCN and are kept in sequence when delivered to the destination.

Finally, it is possible to implement a datagram-like service using a virtual circuit interface. In this case, the exchange of *request* and *accept* packets might be terminated at the subscriber's local packet switch, so that even if packets were not delivered in sequence they might employ abbreviated addressing for local subscriber and packet-switch interaction.

If network interaction is to be based on a standard interface, then agreement must be reached both on the interface and an associated service or services. Furthermore, a common addressing system is needed so that a subscriber on one network can address a packet to a subscriber on any other network. A weaker assumption could be made but we are deliberately assuming a truly common service, interface, and addressing mechanism. We will return to this topic in a later section.

The choice of a standard network service through which to effect network interconnection has a primary impact on the flexibility of implementable network interconnection methods. We will consider two choices: datagram service and virtual circuit service.

a) Datagram service as a standard for network interconnection: For this case, it is assumed that every network offers a common datagram service. A uniform address space makes it possible for subscribers on any network to send packets addressed to any other subscriber on a connected network. Packets are routed between subscriber and gateway and between gateways based on the destination address. No attempt is

made to keep the datagrams in any order in transit or upon delivery to the destination. Individual datagrams may be freely routed through different gateways to recover from failures or to allow load-splitting among parallel gateways joining a pair of networks.

The gateway/gateway interface may be different than the network access interface, if need be (see Fig. 9).

This strategy requires that all networks implement a common interface for subscribers. The simplicity and flexibility of the datagram interface strategy is offset somewhat by the need for all networks to implement the same interface. This is true for the pure virtual circuit interface strategy as well, as will be shown below.

One of the problems which has to be faced with any network interconnection strategy is congestion control at the gateways. If a gateway finds that it is unable to forward a datagram into the next network, it must have a way of rejecting it and quenching the flow of traffic entering the gateway en route into the next network. The quenching would typically take the form of an error or flow control signal passing from one gateway half to another on behalf of the associated network. Similar signals could be passed between subscribers and the packet network for similar reasons. Since datagram service does not undertake to guarantee end/end reliability, it is possible to relieve momentary congestion by discarding datagrams, as a last resort.

b) Virtual circuits for network interconnection: Another alternative standard network service which could be used for network interconnection is virtual circuit service (Fig. 10). Independent of the precise interface used to "set up" the virtual circuit, a number of implementation issues immediately arise if such a service is used as a basis for network interconnection.

Since it is intended that all packets on a virtual circuit be delivered to the destination subscriber in the same sequence as they were entered by the source subscriber, it is necessary that either: 1) all packets belonging to the same virtual circuit take the same path from source subscriber, through one or more gateways, to destination subscriber; or 2) all packets contain sequence numbers which are preserved end-to-end between the source DCE in the originating network and the destination DCE in the terminating network.

In the first case, virtual circuits are set up and anchored to specific gateways so that the sequencing of the virtual circuit service of each network can be used to preserve the packet sequence on delivery. This results in the concatenation of a series of virtual circuits through each gateway and, therefore, the knowledge of each virtual circuit at each gateway (since the next gateway to route the packet through must be fixed for each virtual circuit).

In the second case, there is no need to restrict the choice of gateway routing for each virtual circuit since the destination DCE will have sufficient information to resequence incoming packets prior to delivery to the destination subscriber.

In either case, the destination DCE will have to buffer and resequence packets arriving out of order due either to disordering within the last network or to alternate routing among networks, if this is permitted. Some networks may keep packets in sequence as they transit the network. This will only be advantageous at the destination DCE if the packets enter the network in the desired sequence. If such a service is relied upon in the internet environment, then each gateway must assure that on entry to such a net, the packets are in the desired order for delivery to a destination subscriber or another gateway.

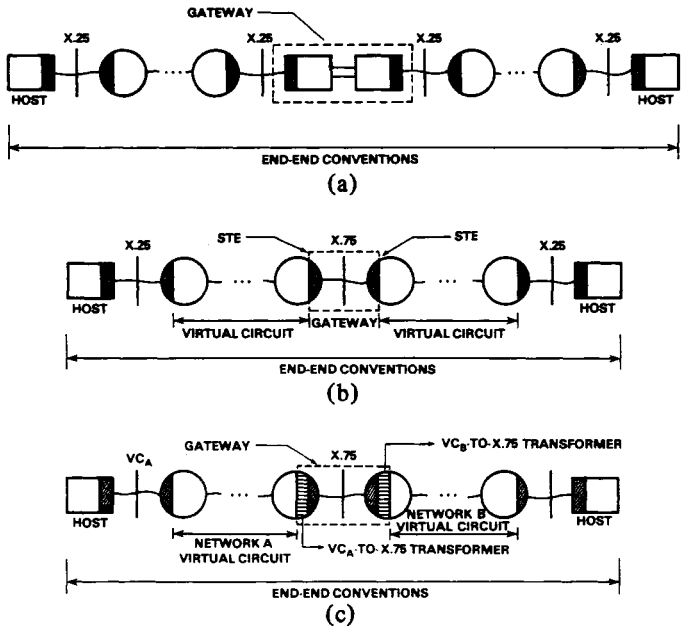

Fig. 10. Virtual circuit network interconnection strategies. (a) Subscriber-based gateway. Internet source and destination carried in user data field of X.25 call set-up packets. (b) X.75 based gateway. Note how much of the X.25 VC service is terminated at the STE. (c) X.75-based gateways with general virtual circuit networks.

The buffering and resequencing of packets within the networks or at gateways introduces substantial variation in buffer space requirements, packet transit delays, and the potential for buffer lockups to occur [50], [51], [61].

If packets for a specific virtual circuit are restricted to pass through a fixed series of gateways, and if a standard flow-control method is agreed upon as part of the virtual circuit service, then it is possible for each internet gateway to participate in end-to-end flow control by modifying the flow control information carried in packets carried end-to-end from the source DCE to the destination DCE. Consequently, a gateway may be able to adjust the amount of traffic passing through it and thereby achieve a kind of internet gateway congestion control. If this is done by allocating buffer space for "outstanding" packets, then either the gateways must guarantee the advertised buffer space or there must be a retransmission capability built into the internet virtual circuit implementation, perhaps between source DCE and destination DCE or between DCE's and gateways.

Such a mechanism does not, however, solve the problem of network congestion unless the gateway-flow control decisions take into account resources both in the gateway and in the rest of the network. Although it is tempting to assume that virtual circuit-flow control can achieve internetwork congestion control, this is by no means clear, and is still the subject of considerable research.

As a general rule, compared to the datagram method, the virtual circuit approach requires more state information in each gateway, since knowledge of each virtual circuit must be maintained along with flow control and routing information. The usual virtual circuit interface is somewhat more complex for subscribers to implement as well, because of the amount of state information which must be shared by the subscriber and the local DCE. For example, implementations of the X.25

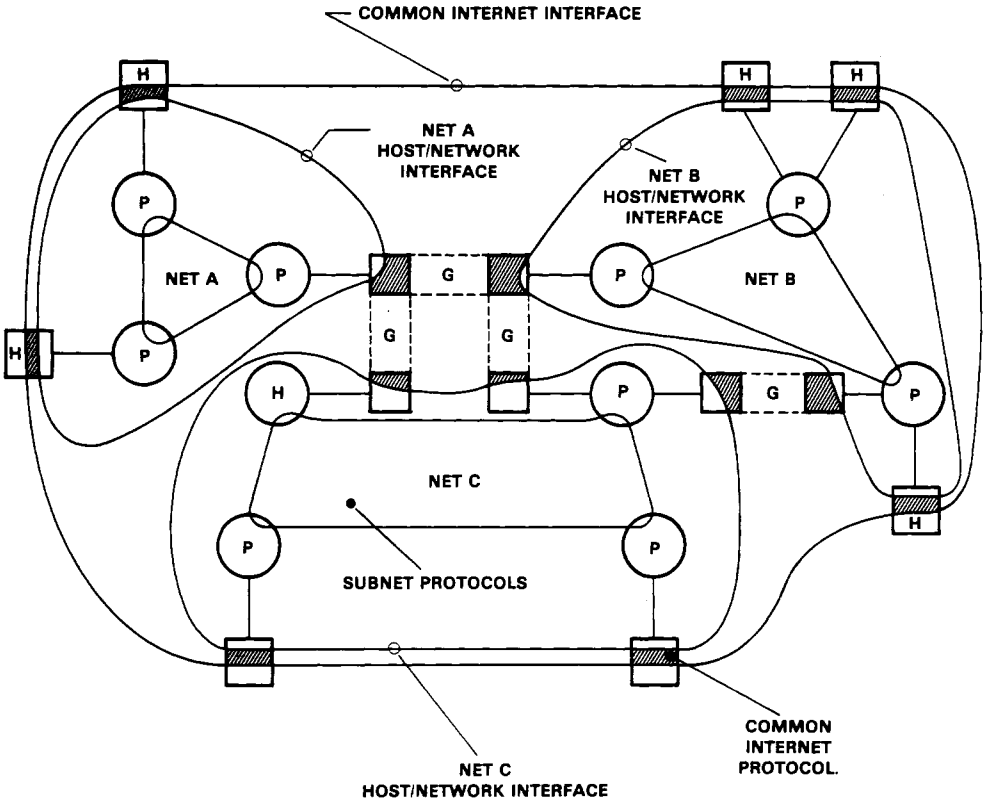

Fig. 11. Common internet interface.

interface protocol have been privately reported by Computer Corporation of America and University College London to require 4000–8000 words of memory on Digital Equipment Corporation PDP-11 computers. By contrast, the ARPANET and Packet Radio Network datagram interfaces require 500–1000 words of memory on the same machine. For internetwork operation, this may be even more burdensome, since any failure at a gateway may require a subscriber-level recovery through an end-to-end protocol, in addition to the virtual circuit interface software, as is shown in [52].

Nevertheless, it may be advantageous to consider internetworking standards which usefully employ both datagram and virtual circuit interfaces and services. For example, some special internet services such as multidestination delivery may be more efficient if they are first set up by control exchanges between the subscriber and the local network and perhaps gateways as well. Once set up, however, a datagram mode of operation may be far more efficient than maintaining virtual circuits for all destinations. Implicit virtual circuits which are activated by simple datagram-like interfaces are also attractive for very simple kinds of terminal equipment.

If it is not possible for all networks to implement a common network-access interface, then the next opportunity is to standardize only the objects which pass from one net to the next and to minimize any requirements for the sequencing of these objects as they move from net to net.

3) General Host Gateways: In this model, a gateway is indistinguishable from any other network host and will implement whatever host/network interface is required by the networks to which it is attached. For many networks, this may be X.25, but the strategy does not rely on this. The principle assumption is that packet networks are at least capable of carrying subscriber packets up to some maximum length, which may vary from network to network. It is specifically not assumed that these packets will be delivered in order through intermediate networks and gateways to the destination host. This minimal type of service is often termed "datagram" service to distinguish it from sequenced virtual circuit service. A detailed discussion of the tradeoff between datagram and virtual circuit types of networks is given elsewhere [52].

The basic model of network interconnection for the datagram host gateway is that internetwork datagrams will be carried to and from hosts and gateways and between gateways by encapsulation of the datagrams in local network packets. Pouzin describes this process generically as "wrapping" [37]. The basic internetwork service is therefore a datagram service rather than a virtual circuit service. The concept is illustrated in Fig. 11.

Datagram service does not offer the subscriber as many facilities as virtual circuit service. For example, not all datagrams are guaranteed to be delivered, nor do those that are delivered have to be delivered in the sequence they were sent. Virtual circuits, on the other hand, do attempt to deliver all packets entered by the source in sequence to the destination. These relaxations allow dynamic routing of datagrams among multiple, internetwork gateways without the need for subscriber intervention or alert.

The internet datagram concept gives subscribers access to a basic internet datagram service while allowing them to build more elaborate end-to-end protocols on top of it. Fig. 12 illustrates a possible protocol hierarchy which could be based on the internet datagram concept. The basic internet datagram service could be used to support transaction protocols or real-time protocols (RTP) such as packet-voice protocols (PVP) which do not require guaranteed or sequenced data

UTILITY	FTP	VTP	RTP	VP
END/END SUBSCRIBER	END/END VIRTUAL CIRCUIT		END/END DATAGRAM	
INTERNET ACCESS	INTERNET DATAGRAM			
NETWORK ACCESS	NETWORK SPECIFIC			
INTRANET, END-END	NETWORK SPECIFIC			
INTRANET, NODE-NODE	NETWORK SPECIFIC			
LINK CONTROL	NETWORK SPECIFIC			

Fig. 12. Protocol layering with internetwork datagrams.

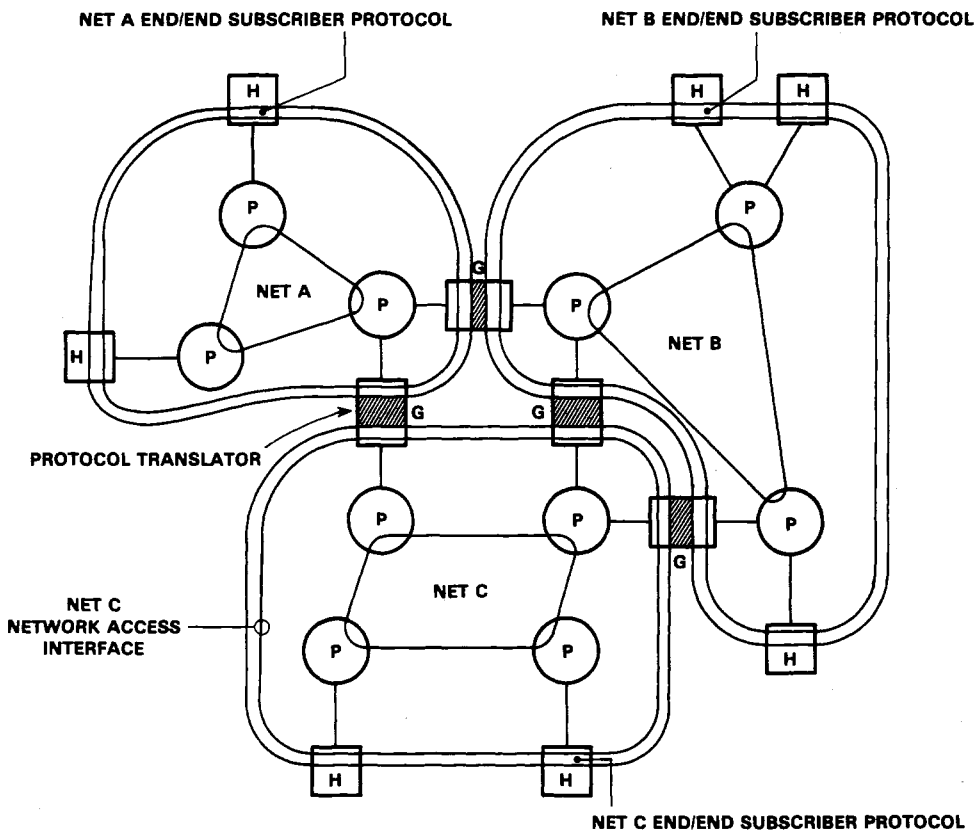

Fig. 13. Host protocol translation gateway.

delivery; reliable, sequenced protocols could be constructed above the basic internet datagram service to perform end/end sequencing and error handling. Applications such as virtual terminal protocols (VTP) [40], [42], [48] or file-transfer protocols [40], [42], [49] could be built above a reliable, point-to-point, end/end service which is itself built atop internet datagrams. Under this strategy, the basic gateway functions are the encapsulation and decapsulation of datagrams, mapping of internet source/destination addresses into local network addresses and datagram routing. Gateways need not have any knowledge of higher level protocols if it is assumed that protocols above the internet datagram layer are held in common by the communicating hosts. Datagrams can be routed freely among gateways and can be delivered out of sequence to the destination host.

The basic advantage of this strategy is that almost any sort of network can participate, whether its internal operation is datagram or virtual circuit oriented. Furthermore, the strategy offers an easy way for new networks to be made "backwards compatible," with older ones while allowing the new ones to employ new internal operations which are innovative or more efficient.

Every subscriber must implement the internet datagram concept for this strategy to work, of course. The same problem arises with the standard network interface strategy since all subscribers must implement the same network interface.

4) Protocol Translation Gateways: It would be misleading to claim that the concept of protocol translation has not played a role in the discussion thus far. In a sense, the encapsulation of internet datagrams in the packet format of each intermediate network is a form of protocol translation. The basic packet carrying service of one network is being translated into the next network's packet carrying service (see Fig. 13). This concept could be extended further. For example, if two networks have a virtual circuit concept, one implemented within the subnetwork and the other through common

host/host protocols, it might be possible, at the gateway between the nets, to map one network's virtual circuit into the other's. This same idea could be applied to higher level protocol mappings as well; for instance, the virtual terminal protocol for one network might be transformed into that of another "on the fly."

The success of such a translation strategy depends in large part on the commonality of concept between the protocols to be translated. Mismatches in concept may require that the service obtained in the concatenated case be a subset of the services obtainable from either of the two services being translated. Extending such translations through several gateways can be difficult, particularly if the protocols being translated do not share a common address space for internetwork sources/destinations. In the extreme, this strategy can result in subscribers "logging in" to the gateway in order to activate the protocols of the next network. Indeed, front-end computers could be considered degenerate translation gateways since they transform host/front-end protocols into network protocols.

There are circumstances when translation cannot be avoided. For instance, when the protocols of one network cannot be modified, but internet service is desired, there may be no alternative but to implement protocol translations. The model typically used to guide protocol translation gateways is that the source/destination hosts lie on either side of the translation gateway. Concatenation of protocol translations through several networks and gateways is conceivable, but may be very difficult in practice and may produce very inefficient service.

C. Names, Addresses, and Routes

In order to manage, control, and support communication among computers on one or more networks, it is essential that conventions be established for identifying the communicators. For purposes of this discussion, we will use the term *host* to refer to all computers which attach to a network at the network-access level of protocol (see Table I). Subscribers to terminal-access services can be thought of as attaching to hosts, even if the host is embedded in the hardware and software of a packet switch as a layer of protocol. Consequently, we can say that the basic task of a packet-switching network is to transport data from a source host to one or more destination hosts.

To accomplish this task, each network needs to know to which destination packets are to be delivered. Even in broadcast nets such as the ETHERNET, this information is necessary so that the destination host can discriminate packets destined for itself from all others heard on the net. At the lowest-protocol levels it is typical to associate destinations with *addresses*. An address may be simply an integer or it may have more internal structure.

At higher levels of protocol, however, it is more common to find text strings such as "MULTICS" or "BBN-TENEX" used as *names* of destinations. Application software, such as electronic mail services, might employ such names along with more refined destination identifiers. For example, one of the authors has an electronic mailbox named "KIRSTEIN at ISI" located in a computer at the University of Southern California's Information Sciences Institute.

Typically, application programs transform names into addresses which can be understood by the packet-switching network. The networks must transform these addresses into *routes* to guide the packets to their destination. Some networks bind addresses to routes in a relatively rigid way (e.g., setting up virtual circuits with fixed routing) while others determine routes as the packets move from switch to switch, choosing alternate routes to bypass failed or congested areas of the network. Broadcast networks need not create routes at all (e.g., SATNET).

In simple terms, a *name* tells what an object is; an *address* tells where it is; and a *route* tells how to get there [54]. A simple model involving these three concepts is that hosts transform names into addresses and networks transform addresses into routes (if necessary). However, this basic model does leave a large number of loose ends. The subject is so filled with issues that it is not possible in this paper to explore them all in depth. In what follows, some of the major issues are raised and some partial resolutions are offered.

One major question is "Which objects in the network should have names? addresses?" Pouzin and Zimmermann offer a number of views on this question in their paper in this issue [37]. A generic answer might be that at least all objects which can be addressed by the network should have names as well so that high-level protocols can refer to them. For example, it might be reasonable for every host connection on the network to have an name and an address. There also may be objects internal to the network which also have addresses such as the statistics-gathering *fake hosts* in the ARPANET [38].

A related issue is whether objects should or can have multiple names, multiple addresses, and multiple routes by which they can be reached. The most general resolution of this issue is to permit multiple names, addresses, and routes to exist for the same object. An example taken from the multinetwork environment may serve to illustrate this notion. Fig. 6 shows three networks which are interconnected by a number of gateways. Each gateway (or pair of gateway halves) has two interfaces, one to each network to which it is attached. Plainly there is the possibility that several alternate routes passing through different gateways and networks could be used to carry packets from a source host in one net to a destination host in another net. This is just the analog of alternate routing within a single network.

Furthermore, each gateway has two addresses, typically one for each attached network. This is just the analog of a host on one network attached to two or more packet switches for reliability. The term *multihoming* is often used to refer to multiply attached hosts.

Finally, it may be useful to permit a gateway to have more than one name, for example, one for each network to which it is attached. This might allow high-level protocols to force packets to be routed in certain ways for diagnostic or other reasons. Multiple naming also allows the use of nicknames for user convenience. Many of these same comments would apply to hosts attached to multiple networks.

An interesting addressing and routing problem arises in mobile packet radio networks. Since hosts are free to move about, the network will need to dynamically change the routes used to reach each host. For robustness, it is also desirable that hosts be able to attach dynamically to different packet radios. Thus failure of a packet radio need not prevent hosts from accessing the network. This requires that host names and perhaps host addresses be decoupled from packet radio addresses. The network must be able to search for hosts or alternatively, hosts must "report-in" to the network so that their addresses can be associated with the attached packet radio to facilitate route selection based on host address. This is just a way of supporting *logical host addressing* rather than using the more common

physical host addressing in which a host's address is an extension of the packet-switch address.

A crucial issue in network interconnection is the extent to which it should or must impact addressing procedures which are idiosyncratic to a particular network. It is advantageous not to require the subscribers on each network to have detailed knowledge of the network address *structure* of all interconnected networks. One possibility is to standardize an internetwork address structure which can be mapped into local network addresses as needed, either by subscribers, by gateways or by both. Subscribers would know how to map internetwork service names into addresses of the form NETWORK/SERVER. Subscribers need not know the fine structure of the SERVER field. Gateways would route packets on the basis of the NETWORK part of the address until reaching a gateway attached to the network identified by NETWORK. At this point, the gateway might interpret the SERVER part of the address, as necessary, to cause the packet to be delivered to the desired host.

The addressing strategy presently under consideration by CCITT (X.121, [30]) is based on the telephone network. Up to 14 digits can be used in an address. The first 4 digits are a "destination network identification code" or DNIC. Some countries are allocated more than one DNIC (the United States has 200). The remaining ten digits may be used to implement a hierarchical addressing structure, much like the one used in the existing telephone network.

Since the CCITT agreements are for international operation, it might be fair to assume that the United States will not need more than 200 public network identifiers. However, this scheme does not take into account the need for addressing private networks. The private networks, under this addressing procedure will most likely appear to be a collection of one or more terminals or host computers on one or more public networks. It is too early to tell how much this asymmetry in addressing between public and private networks will affect private multinetwork protocols.

A related problem which is not unique to network interconnection has to do with addressing (really multiplexing and demultiplexing) at higher protocol levels. The public carriers tend to offer services for terminal as well as host access to network facilities. This typically means that addresses must be assigned to terminals. The issue is whether the terminal address should be associated with or independent of the protocols used to support terminal-to-host communication.

The present numbering scheme would not distinguish between a host address and a terminal address. A host might have many addresses, each corresponding to a process waiting to service calling terminals.

There has been discussion within CCITT concerning "subaddressing" through the use of a user data field carried in virtual call "setup" packets. This notion would support the concept of a single host address with terminal or process level demultiplexing achieved through the use of the user data field subaddressing.

It seems reasonable to predict that, as terminals increase in complexity and capability, it will eventually be attractive to support multiple concurrent associations between the terminal and several remote service facilities. Applications requiring this capability will need terminal multiplexing conventions beyond those currently provided for in the CCITT recommendations.

To simplify implementations of internet protocol software, it is essential to place bounds on the maximum size of the NETWORK/SERVER address. Otherwise, subscribers may have to construct name-to-address mapping tables with arbitrarily large and complex entries.

Even if all these issues are resolved, there is still a question of "source routing" in which a subscriber defines the route to be taken by a particular packet or virtual circuit. Depending on the range of internetwork services available, a subscriber may want to control packet routes. It is not yet clear how such a capability will interact with access control conventions, but this may be a desirable capability if gateways are not able to automatically select routes which match user service requirements.

D. Flow and Congestion Control

For purposes of discussion, we distinguish between flow and congestion control. Flow control is a procedure through which a pair of communicators regulate traffic flowing from source to destination (each direction possibly being dealt with separately). Congestion control is a procedure whereby distributed network resources, such as channel bandwidth, buffer capacity, CPU capacity, and the like are protected from oversubscription by all sources of network traffic. In general, the successful operation of flow-control procedures for every pair of network communicants does not guarantee that the network resources will remain uncongested.

In a single network, the control of flow and congestion is a complex and not well understood problem. In a multinetwork environment it is even more complex, owing to the possible variations in flow and congestion control policies found in each constituent network. For example, some networks may rigidly control the input of packets into the network and explicitly rule out dropping packets as a means of congestion control. At the other extreme, some networks may drop packets as the sole means of congestion control.

At this stage of development, very little is known about the behavior of congestion in multiply interconnected networks. It is clear that some mechanisms will be required which permit gateways and networks to assert control over traffic influx especially when a gateway connects networks of widely varying capacity. This problem is likely to be most visible at gateways joining high speed local networks to long-haul public nets. The peak rates of the local nets might exceed that of the long-haul nets by factors of 30–100 or more. Generic procedures are needed for gateway/network and gateway/gateway flow and congestion control. Such problems also show up in single networks, but are amplified in the multinetwork case.

E. Accounting

Accounting for internetwork traffic is an important problem. The public networks need mechanisms for revenue sharing and subscribers need simple procedures for verifying the accuracy of network-provided accountings.

The public packet-switching networks appear to be converging on procedures which account for subscriber use on the basis of the number of virtual circuits created during the accounting period and the number of packets sent on each virtual circuit. Indeed, it has been argued that accounting on the basis of virtual circuits at gateways requires less overhead than accounting on a pure datagram basis [32]. Scenarios can be cited which support the opposite conclusion.

Suppose there is a choice between setting up virtual circuits for each transaction and sending a datagram for each transaction, and that virtual circuit accounting includes information on each virtual circuit setup (as in the present telephone network). If datagram accounting simply accumulates the number of datagrams sent between particular sources and destinations without regard to the time at which they are sent, then the amount of accounting information which is collected for the datagram case will be substantially less than for the virtual circuit case. In the limit (i.e., one packet per transaction), the virtual circuit accounting information is proportional to $2N$, where N is the number of transactions, while for the datagram case, it is proportional to $\log N$ (base 2). This is simply because the datagram case only sums counts for traffic between source/destination pairs while the virtual circuit accounting would identify start/stop times for each virtual circuit.

Alternatively, if the bulk of the traffic involves a large number of packets per transaction, then the two accounting procedures would accumulate more nearly the same information since each would predominantly involve accounting for packet flow.

If it is chosen not to account for virtual circuit duration, but merely to account independently for the number of virtual circuits and the number of packets sent between source/destination pairs, then the virtual circuit accounting would be closer to the datagram case.

The important conclusion to be drawn is that accounting for datagrams is generally less complex than accounting for virtual circuits, but that the two can be made arbitrarily similar by suitable choice of the details of the accounting information collected.

F. Access Control

In multinetwork environments, it may be necessary for each network to establish and enforce a policy for "out-of-network" routing. For example, a public network might conclude agreements with other networks regarding the type and quantity of traffic it will forward into other networks. This might even be a function of the time-of-day. Consequently, mechanisms are needed which will permit networks to prevent traffic from entering or leaving or to meter the type and rate of traffic passing into or out of the network.

Another example of the need for control arises with the possibility of third-party routing. That is, traffic destined from network A to network B is routed through network C. It cannot be assumed that all networks have gateways to all others. However, some nets may want to limit the amount of *transit* traffic they carry. There may be explicit agreements among a subset of the nets regarding revenue sharing for transit services. If a particular network does not have a revenue-sharing agreement with the particular source/destination networks of a given virtual circuit or datagram, then it must be able to reject the offending traffic if it so chooses.

There does not seem to be any technical barrier to separating the access control policy decision mechanism from the enforcement of the policy. For example, a gateway might simply enforce policy by sending traffic for which it has no known access rules to an *access controller.* If we adhere to the model that gateways have two *halves*, then each half deals with the network to which it is connected. The access controller can either dynamically enable the flow by causing table entries at the gateways which permit the flow to be created or it can tell the gateway to reject all further traffic of that type.

Clearly, access control policies will affect routing strategies, so this adds a complicating factor into any internetwork routing strategy implemented by the gateways. At present, very little experience has been accumulated with internet access control and routing policies. For the most part, agreements among public networks have been bilateral and transit routing has been treated as a very special case. When EURONET [6] becomes operational, this problem will be particularly important to solve.

G. Internet Services

It is by no means clear what set of services should be standardized and available from, at least, all public data networks. The current CCITT recommendations provide for virtual circuit service and terminal access service on all public packet-switching networks.

Although the recommendations (X.3, X.25) provide for *fragmentation* of packets being delivered to a subscriber on a virtual circuit, the current X.75 gateway draft recommendation uses an agreed maximum packet size of 128 octets of data, not including the header. This agreement avoids for the moment the need to fragment packets crossing a network boundary, as long as all subscribers recognize that the maximum length internetwork packet allowed is 128 octets. Bilateral exceptions to this rule may develop but neither a fixed size nor a collection of special cases represent a very general solution to this problem.

It has been argued [25] that a general scheme for dealing with fragmentation is desirable so that new network technologies supporting larger packet sizes can be easily integrated into the multinetwork environment.

Apart from fragmentation, there are a set of special services such as multidestination addressing and broadcasting which could be used to good advantage to support multinetwork applications such as teleconferencing, electronic mail distribution, distributed file systems, and real-time data collection. Other services such as low delay, high reliability, high bandwidth, and high priority are also candidates for standardization at the internet level.

As in the case of access control, selection of such services might constrain the choice of packet routing to networks capable of supporting the desired services. Once again, very little experience with standard internet services has been accumulated so this subject is still a topic for research. For the most part, terminal-to-host services have been successfully offered across network boundaries using nearly all of the network interconnection methods described in this paper. It remains to be seen whether more complex applications can be equally well supported.

VII. X.25/X.75—The CCITT Strategy for Network Interconnection

The common network access interface concept is favored by CCITT for network interconnection. In the CCITT model of packet networking, all networks offer the same interface to packet-mode subscribers and this is called X.25. X.25 is a virtual circuit interface protocol. However, gateways between networks employ an interface protocol called X.75 [33], which is much like X.25 but accommodates special network/network information exchange, such as routing information, accounting information, and so on.

Fig. 10(a) illustrates the basic network interconnection strategy proposed by CCITT. To appreciate the difference

between this strategy and the "common subnetwork" strategy, it is necessary to have some understanding of the X.25 packet network interface. X.25 provides a virtual circuit interface for the setup, use, and termination of virtual circuits between subscribers of the networks. X.25 provides for flow control of packets per virtual circuit flowing into or out of the network. Subscribers may set up switched virtual circuits by sending "call request" packets into the network and receiving "call confirmation" packets in return. The standard also provides for permanent virtual circuits.

The public networks plan to employ X.25 interfaces; it can therefore be assumed that source and destination hosts in different networks will essentially want to exchange "call request" and "call accepted" packets through the mediation of one or more gateways. This strategy could result in a series of virtual circuits chaining source host to gateway, gateway to gateway, and gateway to destination host; alternately an end-to-end virtual circuit could be set up from source host to destination host, with the gateways acting as relays without any special knowledge of the virtual circuits passing across the network boundary.

The principle difference between the X.25 interface and X.75 interface is that virtual circuit setup and clearing packets are passed transparently by the X.75 gateway to the next gateway or destination. For reasons which are described below, it is necessary to maintain the sequence of packets belonging to a given X.25 virtual circuit as they pass through a gateway and enter the next network. Therefore, a virtual circuit is in fact created between the source host and intermediate gateway and between gateways. The X.75 gateway does not spontaneously generate any "call acceptance" packets in response to "call request" packets, but it does participate in the sequencing and flow control of packets on each virtual circuit passing through. Other differences between the X.25 and X.75 interface have to do with the nature of the internetwork accounting or routing information which might be exchanged over X.75 which would not be appropriate for a subscriber to exchange with the network over the X.25 interface.

The design of the X.75 type of gateway depends in principle upon all networks' use of the X.25 subscriber interface. Some networks, like the ETHERNET, cannot implement it without extensive modification, because there are no packet switches in the network to support the required packet reordering at the destination. The alternative is to insist that all internet applications rely on a sequenced data protocol built into the hosts or front-ends. For some services, such as packet speech, the potential overhead of resequencing packets before delivery to the destination may prevent the service from being viable. This problem could be amplified if packets are constrained to remain in sequence as they pass the X.75 boundary.

Fig. 10(b) and (c) shows variants of the CCITT interconnection strategy. In Fig. 10(b), we see an example in which only X.25 is used both as a network access method and as a means of passing traffic across network boundaries. A single subscriber or a pair of subscribers to two nets could interface to their networks via X.25 and to each other by means of some agreed and possibly private protocol.

Virtual circuits would be explicitly set up from source host to gateway, gateway to gateway, and gateway to destination host. The "internet" addresses of the source and destination hosts could be carried in the so-called "Call User Data Field" of an X.25 Call Request packet. This leaves the packet address field free to identify intermediate destinations (e.g., gateways), but preserves an ultimate internetwork source/destination address which the gateway can use to select the destination to which the next intermediate virtual circuit is to be set up.

An alternative to this is shown in Fig. 10(c) in which the subnets *A* and *B* use nonstandard virtual circuit interfaces, but agree to build gateway software employing X.75 signaling procedures across the gateway interface. This solution is substantially the same as that shown in Fig. 10(b), except there is now additional translation software in each gateway half to make each virtual circuit network-access protocol compatible with X.75 procedures.

There are some specific problems with the X.25/X.75 gateway strategy, which do not necessarily apply to other virtual call gateways [63]. The basic X.25 interface provides for the sequence numbering of subscriber packets mod. 8 or, optionally, mod. 128. Since X.25 is an interface specification, this numbering can only be relied upon to have local significance (i.e., host-to-packet switch). Some X.25 implementations use these host-assigned sequence numbers on an end-to-end basis. Others generate internal, network-supplied numbers to allow for repacking of subscriber packets into larger or smaller units for transport to the destination. If packet sequence numbers assigned by the source host were carried transparently to the destination without change, it might be possible to allow packets to flow out-of-order across the X.75 boundary to a gateway and thence into the next network. If the packet sequence numbers were still intact, they could be carried out-of-order to the next destination which might either be a gateway or an X.25 host. In the latter case, the original packet-sequence numbers could be used to resequence the packets before delivery. If the packets were being delivered to an intermediate gateway, they would not have to be sequenced there. However, the X.25 interface specification does not undertake to carry the host-supplied sequence numbers to the destination gateway or host in a transparent fashion, primarily so that the subnetwork can deal more freely with the physical packaging of the packet stream. For example, a source may supply packets of length 128 bytes while a destination may prefer to receive packets no longer than 64 bytes. To allow for such variations, the network must be free to renumber packets for delivery. These considerations have two consequences.

1) X.25 packet sequence numbers cannot be relied on for end-to-end signaling, though they could be so used if requisite information is known about the intermediate transit networks.

2) Packets must be delivered in sequence when passing to or from gateways and hosts on X.25 networks.

The second conclusion may be modified slightly. It is at least essential that packets be delivered in relative sequence on each virtual circuit. By maintaining independent sequence numbering on each virtual circuit, it is possible for hosts and gateways to refuse traffic on one virtual circuit while accepting traffic on another. There are two penalties for this. First, a gateway must keep track of which virtual circuits are passing through it. Second, dynamic alternate routing of packets belonging to the same virtual circuit through alternate gateways is not possible without resetting or clearing the virtual circuit. This last point is simply the consequence of not defining an end-to-end sequence numbering scheme, but instead relying on sequencing of the packets of a virtual circuit on entry to and exit from each intermediate network.

Some networks implement X.25 level acknowledgments (i.e., level 3) that have an end-to-end significance, but others make this purely a host-to-packet switch matter. As a conse-

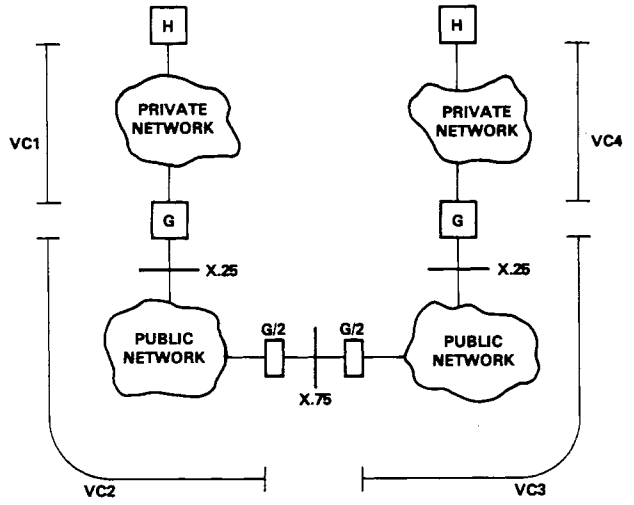

Fig. 14. Use of X.25 for public/private network interconnection.

quence, it is not possible to rely on X.25 packet acknowledgments to determine which, if any, packets were not delivered as a result of the resetting or clearing of a virtual circuit. Furthermore, even if a subnet were to offer an end-to-end acknowledgment between a source host and an X.75 gateway, this could not be assumed to guarantee that the acknowledged packet was delivered to the ultimate X.25 destination in another network.

X.75 is an interface intended for use between public networks. Thus, it is not likely to be used or even allowed as an interface between public networks and private networks. For the case illustrated in Fig. 14, X.25 interfaces could be provided between public and private networks (or other special interfaces) and X.75 interfaces between public networks. Consequently, gateways between public and private networks are likely to appear to be ordinary host computers in the view of the public networks.

The use of X.25 for private/public network interfaces and X.75 for public/public network interfaces leads to the situation shown in Fig. 14 in which an internetwork virtual circuit would have to be made up of several concatenated parts such as virtual circuits 1-2-3-4 (see also [52, Fig. 3.4]). Even if X.25 implementations uniformly permitted an end-to-end interpretation of packet sequence numbers and acknowledgments, there would still be separate virtual circuits required between the source or destination hosts and the gateways into the public networks. However, the concatenation of virtual circuits does not yield a virtual circuit. For instance, a gateway between the public and private net could acknowledge a packet but fail to get it delivered, in which case the subscriber will have been misinformed as to the delivery of the packet. This situation forces the end subscribers of private networks to implement end-to-end procedures on top of any concatenated virtual circuits provided by the public networks.

VIII. Practical Network Connections and Experiments in Progress

A number of networks have been connected successfully over the last few years. Most of these connections have been made in an *ad hoc* manner, using one of the following techniques.

1) One network is a star network with remote RJE and interactive stations. The other is a star or distributed network with clearly defined protocols. A device on the star network provides exactly the functions required by its own network on one side, and those of the other network on the other side.

2) Formal gateways are provided between the two networks, and protocol mapping occurs in the gateway.

3) A computer is a host on two networks. It is arranged that services are provided by accepting input from one network and putting it out on another, possibly after substantial processing.

4) Formal gateways are provided between the two networks. Sufficient agreement is obtained that end-to-end protocols (even high level ones) are common in the two networks. In this case, less activity is required in the gateway.

In the first method, a form of front-end computer is used. It has been adopted in the large airline and banking networks SITA [13] and SWIFT [14]. In each case the standards for the networks have been defined rigidly. SWIFT has even certified officially the devices of three manufacturers to provide interfaces to its network. The other side of the device is then programmed to meet the requirements of the star system being attached. In the two cases cited, only a simple message level of interface needed to be defined.

Other examples of the same technique are the connection of the Rutherford Laboratory (RL) star system [53] and the Livermore CTRNET to ARPANET. In these examples, more serious protocol mapping was required. ARPANET has a well-defined set of HOST-IMP, HOST-HOST, Virtual Terminal, and File Transfer protocols. All these had to be mapped into the appropriate procedures for the other network.

The second method has been applied only experimentally. The UCL interface between ARPANET and the UK Post Office Experimental Packet Switched Service (EPSS, [55]) and the National Physical Laboratory interface between EPSS and the European Informatics Network (EIN, [56]) are examples of this technique; a demonstration has even been made of EIN-EPSS-ARPANET with no extra problems encountered from the three networks being concatenated. Technically there is almost no difference between the first two methods. The second looks at first sight somewhat more general than the first, but almost the same problems have to be overcome. The difficulties come from the fundamental differences in the design choices made in the protocols of the different networks; these differences are in general difficult, and even sometimes impossible, to resolve completely. In the first method, they can sometimes be resolved using a specific facility in the star network; in the second, where two distributed networks are involved, this recourse may no longer be available.

One example of the problem occurs in the connection of EPSS and ARPANET. ARPANET can forward any number of characters at a time, and often uses full duplex remote echoing. EPSS works in a half-duplex mode, forwarding only complete records. A special "Transmit Now" has to be input by the user, and interpreted by the gateway, to ensure that partial records are forwarded. Another example, from the same application, occurs in File Transfer. ARPANET assumes an interactive process is live throughout the file transfer; all completion codes are passed over this live channel. The RL network (and EPSS) assume that file transfer is a batch process; they return network completion codes at a later time, and may delay acting on the commands. With the ARPANET-RL link [53], the file transfer job had to be given a very high priority, so that the completion code usually arrived before a timeout occurred; because of the nature of the way the computer was

used for large real-time jobs, this did not always ensure that the job was run in a reasonable time.

There are several examples of the third technique. A DEC PDP 10 machine used on the Stanford University SUMEX project is a host both on ARPANET and on TYMNET; several machines at Bolt, Beranek and Newman are both on ARPANET and TELENET. Because the TENEX operating system has good facilities for linking between programs, it would be possible for interactive streams to come in one network and go out on another. File transfer problems would be simple in this configuration, because the hosts obey all the conventions, in any case, of each network. Of course, this mode of operation may require that files in transit between networks may have to be stored temporarily in their entirety in the host serving as the gateway between the networks.

The fourth technique is newer, and has many variations. As a result of agreement on the X.25, and partial agreement on the X.75, protocols, PTT networks are able to interconnect in a reasonably straightforward manner. The connections between DATAPAC and both TELENET and TYMNET have been done in this way. In each case, there has not been any agreement on higher level protocols, so the problems of host-host communication across concatenated networks is not resolved by these linkups of the subnets.

The ARPA-sponsored INTERNET project has tried to standardize to a higher level. A host-host protocol has been defined (TCP, [25]), and is being implemented on a number of different networks including Packet Radio [20], [21], ETHERNET [18], LCSNET [64] and the SATNET [22], in addition to ARPANET. This protocol is defined for use across networks; thus each packet includes an "Internet Header" which is kept invariant as the packet crosses the different networks. One aspect of the INTERNET program is to develop gateways which can interpret this header appropriately.

By late 1976, the ARPA project had connected together the Packet Radio Network, the ARPANET, and the Atlantic Packet Satellite Network using two gateways between the Packet Radio Network and the ARPANET and three gateways between the ARPANET and Packet Satellite Network. It is routinely possible to access ARPANET computing resources via either of the other nets and to artificially route traffic through multiple nets to test the impact on performance. In one such test, a user in a mobile van in the San Francisco area accessed a DEC PDP-10 TENEX system at the University of Southern California's Information Sciences Institute over the following path:

1) from van to the first gateway into ARPANET via the Packet Radio Network;
2) across the ARPANET to a second gateway in London, using a satellite link internal to the ARPANET;
3) across the Atlantic Satellite Network to a third gateway in Boston;
4) across the ARPANET again to USC-ISI.

The user and server were 400 geographical miles apart, but the communication path was 50000 miles long and passed through three gateways and four networks. Except for a slightly increased round-trip delay time, service was equivalent to a direct path through the ARPANET. Since the Packet Radio Network is potentially lossy, can duplicate packets, and can deliver packets out of order, the end/end TCP protocol was used to exercise flow and error control on an end-to-end basis. The availability of a common set of host-level protocols substantially aided the ease with which this test could be conducted.

The ARPA project also has high-level standard protocols already in existence to support file transfer and virtual terminals (the FTP and TELNET protocols [40]), and these are being retrofitted above the internet TCP protocol to provide a standard high-level internetwork protocol hierarchy.

IX. Regulatory Issues

The regulatory issues in the interconnection of packet networks takes a different form in North America than elsewhere. It is hard in a paper of this type to more than touch on some of the problems involved. The discussion here is simplistic in the extreme, and no attempt is made to put the issues in the legalistic language they really require.

In almost all countries the provision of long distance communication transmission and switching is provided by a regulated carrier. In most countries outside North America, this carrier is a single national entity—called the "PTT". In some countries (e.g., Italy) there are different carriers for different services—e.g., telegraph, telephone, intercity, international telephone, etc. In North America there are many carriers. Usually only one in each geographical area has a monopoly on public switched voice traffic. Also the so-called "Record Carriers" have some sort of monopoly on "record traffic," which is message traffic. In a "Value Added Network" (VAN), the operators rent transmission equipment from the carriers, and then add their own switching equipment. These VAN's are themselves regulated in what they may do, what traffic they may carry, and what rates they may charge. Between North America and Europe, specific "International Record Carriers" (IRC) have monopoly rights on data and message transmission —in collaboration with the appropriate European PTT's. The regulations take into account who owns the hosts and terminals, who owns the switches, who rents the transmission lines, what types of traffic is carried, what is the geographic extent of the network, and what is the technology of long distance transmission.

In Fig. 15, a single network N is sketched. It consists of switches S and transmission lines L; these together are called the data network, DN. It consists also of terminals T and hosts H; the exact difference between a terminal and a host is not very clear; we believe it is assumed that terminals mainly enter and retrieve data without processing; while a host transforms the information by processing. This definition probably does not meet the picture of modern "intelligent terminals," but it is always hard for the regulations to keep up with the technology. If the total network is all localized in one site, so that no communication lines cross public rights of way, then it can usually be considered from a regulatory viewpoint, as a single host in more complex network connections. The hosts and the terminals can be connected to the switches, and the switches to each other, either by leased lines, or by the Public Switched Telephone Network; the first type of connection is called a *leased* connection, the second *switched*. In the subsequent discussion of this section, the term "host" will include localized networks. In general we will assume the connections between the switches are via leased lines; if that is not the case, the regulations are much eased in general (though in some countries, like Brazil, no data transmission is permitted at all via switched telephone lines).

If all the hosts and switches are owned by one organization P, which also leases the lines, then P is said to own and operate the network, and it is called a "Private Network." There are

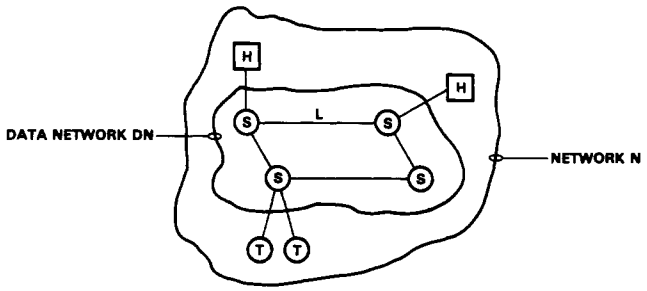

Fig. 15. Schematic of one network.

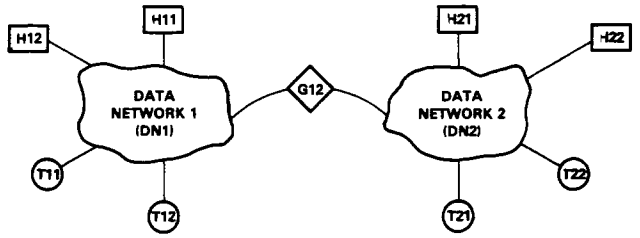

Fig. 16. Schematic of two connected networks.

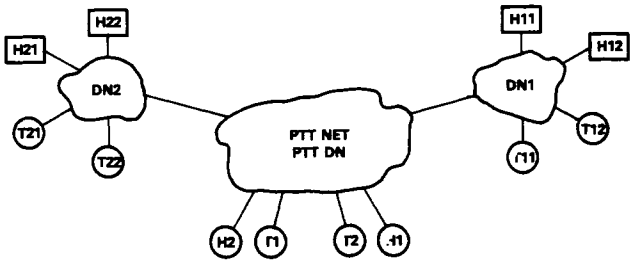

Fig. 17. Schematic of PTT model.

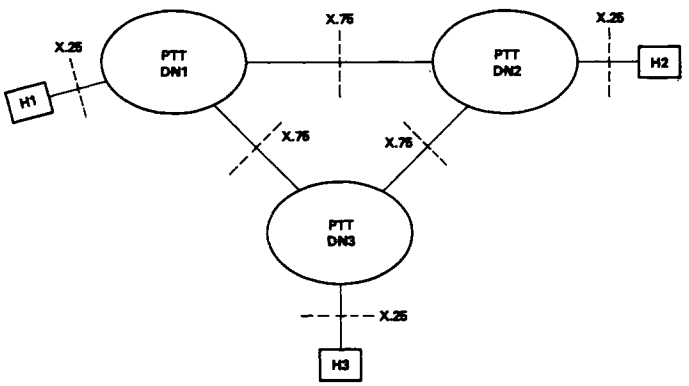

Fig. 18. Multiple PTT network interconnection.

minimal restrictions on such networks—though in West Germany, for example, higher tariffs are charged for the leased lines if any terminals or hosts are connected via the PSTN. In most countries such a network may not be used for the transfer of messages between terminals belonging to organizations other than *P*.

If the data network belongs to one organization, and the hosts to others, the data network is a VAN. Stringent regulations apply to VAN's, in most countries. With rare exceptions, in most European countries, VAN's can be operated only by the PTT's. In the U.S., they can be operated by other organizations, but only if approved as regulated Value Added Carriers (VAC's) by the Federal Communications Commission (FCC). One regulation imposed by the U.S. is that an organization operating as a VAC may not also operate a host for outside sale of services. For this reason, the companies TYMSHARE and ITT have had to spin off their VAC's into separate subsidiaries, TYMNET and ITT Data Services.

In the past, a few VAN's have been permitted to operate internationally for specific interest groups. Two such VAN's are SITA [14], for the airlines, and SWIFT [14] for the banking community. Here the regulations can be stringent. SWIFT has to pay specially high tariffs for its leased lines; its license to operate may be revoked when the PTT's can offer a comparable international service.

As soon as two networks, owned by different organizations, are interconnected, there are regulatory difficulties. This situation is illustrated schematically in Fig. 16. Even if one network is an internal one, so that it can be treated as a single host, its connection to other network immediately changes the latter's status. Thus in Fig. 16, the connection of *DN*1 to *DN*2 immediately changes *DN*2 to a VAN. In Europe it has been decreed that such private networks may not connect directly to each other, but only through a PTT network. Thus the most general configuration permitted by the European PTT's is illustrated in Fig. 17. Moreover, the PTT's have also agreed that only the X.25 interface will be provided to customers, though that interface was defined for the configuration of Fig. 15 rather than 17. The different PTT networks will themselves connect to each other by the different interface X.75 as illustrated in Fig. 18. This does not change, however, the interface seen by the private networks. Further work is needed to assess the suitability of X.25 in this role.

In the U.S., the regulations are not quite so stringent. Connections such as Fig. 15 are permitted even where one host belongs to a different organization than the network operator *P*—provided such connection is only limited and for the purposes of using the facilities of that network. This type of relaxation is really necessary, because of the difficulty of distinguishing between a "host" and a "terminal". In practice, in most countries, the line is drawn between leased line and PSTN connections. The former are usually not permitted without change of status of the network; the latter seem to downgrade the connection to that of a terminal.

The discussion above has treated the types of connections which can be made. In addition, the PTT's, and the FCC in the U.S., usually regulate the purposes for which the network can be used. In particular, there is a ban on such networks being used for message or voice transmission between organizations. How such measures are to be policed, gets us into another regulatory problem. For example the UK PO [57] has claimed a right to inspect the contents of any data message sent across lines leased from it; this right would be at variance with the privacy laws being enacted in many countries [58], [59]. This subject is a large one in its own right, and it is clearly beyond the scope of this paper.

Two other service problems will arise in international connections. First the impact and form of the privacy and transnational data flow regulations in different countries are different. Thus in the interconnection of international networks, a particular set of problems may arise, even when the appropriate regulations are obeyed in each network separately. Thus both Network 1 in country *A* and Network 2 in country *B* may obey their own national regulations. However when the

networks *A* and *B* are connected, Network 1's practices may break country *B*'s regulations, and yet be accessible from country *B*. It is this class of problems which delayed seriously the permission by the Swedish Data Inspectorate Board for Swedish banks to connect their networks to SWIFT.

Secondly, some of the functions of networks or gateways legal in one country may be illegal in another. Thus U.S. carriers are not permitted to do data processing in their data networks; no such considerations apply in most European countries. Some of the protocol translation activities, some of the message processing activities, and some of the high-level services (e.g. the provision of multiaddress links) may well be classed as "Data Processing," and hence be illegal in the U.S. In interconnected networks, this raises the possibility that functions can be carried out outside the jurisdiction of the country in which the operator initiating the activity is sited, and yet which is illegal in that country. This subject is treated rather fully elsewhere [60]. A clear example of this is the use of message services operated by TYMSHARE and CCA on TELENET and TYMNET. While these services are legal in the U.S., their use by UK persons connected to TYMNET by the official International Packet Switched Service is clearly technically illegal; this use would contravene the UK Post Office Monopoly.

X. Unresolved Research Questions

There are many unresolved research questions; on some of them even the present authors do not agree with each other! Primarily these questions have a technical, policy, administrative, economic, regulatory, or operational aspect, or a combination of these.

One example of this is the question of the procedures to be used for internet routing. Here there are technical questions on what is feasible in view of the technologies used in the subnets; there are policy questions on when third country routing might be allowed; there are economic considerations on how much it would cost to do the necessary protocol translation to route through third countries, and on what charges the connecting transit network might make; there may be regulatory questions on which classes of data may flow through specific countries (related to the transnational data flow regulations); and there may be operational questions on whether in the event of failure in dynamic rerouting, reestablishment could take place with sufficient rapidity.

Among the outstanding research questions are, in alphabetic order, the following.

Access Control: What are the requirements and methods of implementation of access control? How should they affect internetwork routing?

Addressing: How should the International Numbering Plan, which goes to the level of known subscribers of public networks, be extended? Should this extension be in the numbering plan itself, or should additional user and network information be supplied? Should there be local, or only physical, addressing? Should there be internetwork source routing implied by the addressing?

Broadcast Facilities: What is the role of broadcast communication facilities in the provision of internet services? Should facilities using it be offered? Should technologies supporting it use it, particularly at gateways? What are the implications on protocols, especially with respect to duplicate and error detection?

Datagram versus Virtual Call Facilities: How should datagram and virtual call facilities be interconnected? How can one compare the relative performance and costs of the implementations? What criteria should be used in any comparison? When might datagram, or alternatively virtual calls, be desirable or essential between networks?

Data Protection: What are the effects of end-to-end data encryption on protocol translation?

Flow and Congestion Control: To what extent should one adopt congestion and flow control between gateways and their feeding networks, between gateways directly, or between gateways and the source? What are the relative effects of just discarding packets in gateways, and relying on the end-to-end protocol to detect and compensate for this? How is charging for discarded packets arranged?

High Level Protocols: There are still many questions on what should be standardized, and how rigid the standards should be. To what extent should the individual networks support common standards, and to what extent should protocol translation be feasible technically or attractive economically? What are the costs of maintaining standards or the economic advantages of standard hardware and software? How does the technology of individual networks and the proportion of internetwork traffic affect the decisions?

Internetwork Diagnosis: There are many technical problems in isolating faults in concatenated networks. There are also organizational and economic problems on who should be responsible for their repair, and how costs for service failures should be allocated.

Performance: How do choices of design parameters, and network services, affect the costs of the individual networks? How do the individual network performances and costs scale to large networks? How do the choices affect the feasibility, costs and performance of the gateways? How do the variations in technology or choice of parameters affect the performance in interconnected networks?

Routing Policies: To what extent and when should adaptive routing be used between networks? How can one recover from the partitioning of a single network, when there are still routes existing by going through other networks? How should administrative considerations affect routing policies between networks (privacy regulations, economic considerations of internet payments, desire to provide for high availability, etc.)? When is a hierarchical organization more efficient that a direct route search?

Services: What services are needed on an internetwork level? Clearly interactive and bulk transport services must be supported. What else is needed? Should the internetwork facilities be able to support voice, telemetry, and teleconferencing? What is the cost of supporting these latter services, and what is their effect on other facilities?

X.25 and X.75 and Related Recommendations: Is X.25 suitable for transaction processing? Are the present datagram proposals adequate? How should X.25 be extended for internet addressing? How should X.25/X.75 be modified to allow the connection of private to public networks, or private networks to each other? Do the X.3, X.28, X.29 pad concepts extend well to the internet environment, or should they be modified?

XI. Conclusions

In view of all the unresolved questions discussed in Section X, most of the conclusions which can be drawn in this paper must be tentative. From the early part of the paper, we have shown that it is essential that techniques be developed for con-

necting computer networks. Moreover, no single set of techniques will fit all applications.

The services which will normally have to be supported are terminal access, bulk transfer, remote job entry, and transaction processing. The quality and facilities of the services required will be very dependent on the applications.

The connections between networks can be made at the level of the packet switches or of hosts, and can be on a datagram or virtual call basis. Connection at the packet-switch level requires broadly similar network access procedures, or complex protocol transformation at the gateways between the networks. If the network protocols are different, interconnection can be most easily achieved if done at the host level. The higher levels of service can be mapped at service centers, which need not be colocated with the gateways—but very different philosophies of network services can be very difficult to map. Alternatively, subscribers can implement common higher level protocols if these can be agreed upon.

The principal problems in connecting networks are much the same as those in the design of the individual networks of heterogeneous systems—but the lack of a single controlling authority can make the multinet design problem more difficult to solve. It is essential to resolve the usual problems of flow control, congestion control, routing, addressing, fault recovery, flexibility, protocol standards, and economy. The public carriers have attempted to resolve many of these problems; particularly in the areas of flexibility, addressing, and economy we feel their solutions are not yet adequate. At the higher levels of protocol, much more standardization is required before we have really satisfactory long term solutions.

The advent of international computer networks, private networks which must communicate with other private networks (even if via public ones), and the new applications of computer networks, raise regulatory and legal issues which are far from resolution.

Many technical solutions to the problems of the connection of networks are discussed in this paper. Their applicability in view of the different technical, economic, and policy constraints imposed in different countries must still be assessed.

REFERENCES

[1] L. G. Roberts, "Telenet: Principles and practice," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, pp. 315-329, 1975.

[2] W. W. Clipsham, F. E. Glave, and M. L. Narraway, "Datapac network overview," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 131-136, 1976.

[3] J. Rinde, "TYMNET: An alternative to packet switching technology," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 268-273, 1976.

[4] R. E. Millstein, "The national software works: a distributed processing system," in *Proc. ACM Nat. Conf.*, Seattle, WA. 1977.

[5] A. Danet, R. Despres, A. Le Rest, G. Pichon, and S. Ritzenthaler, "The French public packet switching service, The TRANSPAC network," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 251-269, 1976.

[6] G. W. P. Davies, "EURONET project," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 229-239, 1976.

[7] R. Nakamura, F. Ishino, M. Sasaoka, and M. Nakamura, "Some design aspects of a public switched network," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 317-322, 1976.

[8] F. A. Helsel and A. J. Spadafora, "Siemens system EDS—A new stored program controlled switching system for telex and data networks," in *Proc. Third Int. Computer Communications Conf.*, Toronto, Canada, pp. 51-55, 1976.

[9] T. Larsson, "A public data network in the nordic countries," in *Proc. Third Int. Computer Communications Conf.*, Toronto, Canada, pp. 246-250, 1976.

[10] P. T. Kirstein, "Planned new public data networks," *Comput. Networks*, vol. 1, no. 2, pp. 79-94, 1976.

[11] P. T. F. Kelly, "An overview of recent developments in common user data communications networks," in *Proc. Third Int. Computer Communications Conf.*, Toronto, Canada, pp. 5-10, 1976.

[12] ——, "Public packet switched data networks," this issue, pp. 1539-1549.

[13] P. Hirsch, "SITA rating a packet-switched network," *Datamation*, vol. 20, pp. 60-63, 1974.

[14] G. Lapidus, "SWIFT network," *Data Communications*, vol. 5, no. 5, pp. 20-24, 1976.

[15] L. G. Roberts and B. D. Wessler, "The ARPA network," in *Computer-Communications Networks*, N. Abramson and F. Kuo, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1973, pp. 485-500.

[16] P. M. Karp, "Origin, development, and current status of the ARPANET," in *Proc. COMPCON73*, San Francisco, CA, Feb.-Mar. 1973, pp. 49-52.

[17] L. Pouzin, "Presentation and major design aspects of the CYCLADES computer network," in *Proc. Third Data Communications Symp.*, Tampa, FL, Nov. 1973, pp. 80-85.

[18] R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," *Commun. ACM*, vol. 19, no. 7, pp. 395-404, July 1976.

[19] A. S. Fraser, "SPYDER—A data communications experiment," *Comput. Sci. Tech. Report*, no. 23, Bell Laboratories, Dec. 1974.

[20] R. E. Kahn, "The organization of computer resources in a packet radio network," in *Proc. Nat. Computer Conf.*, AFIPS Press, pp. 177-186, May 1975.

[21] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," this issue, pp. 1468-1496.

[22] I. M. Jacobs, R. Binder, and E. V. Hoversten, "General purpose satellite networks," this issue, pp. 1448-1467.

[23] D. Lloyd and P. T. Kirstein, "Alternate approaches to the connection of computer networks," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 499-504, 1975.

[24] L. Pouzin, "A proposal for interconnecting packet switching networks," IFIP Working Group 6.1, General Note no. 60, Mar. 1974.

[25] V. G. Cerf and R. E. Kahn, "A protocol for packet network interconnection," *IEEE Trans. Commun. Technol.*, vol. COM-22, pp. 637-641, 1974.

[26] C. Sunshine, "Interconnection of computer networks," *Comput. Networks*, vol. 1, 1977, pp. 175-195.

[27] CCITT, "Recommendation X.3: International user facilities in public data networks," *Public Data Networks, Orange Book*, vol. viii.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 21-23, 1977.

[28] CCITT, "Recommendation X.25: Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DEC) for terminals operating in the packet mode on public data networks," *Public Data Networks, Orange Book*, vol. VIII.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 70-108, 1977.

[29] CCITT, "Provisional recommendations X.3, X.25, X.28 and X.29 on packet-switched data transmission services," Int. Telecommunications Union, Geneva, Switzerland, 1977.

[30] CCITT, "Recommendation X.121—Int. numbering plan for public data networks," Study Group VII, Temporary Document 76-E, Int. Telecommunications Union, Geneva, Switzerland, April 25, 1978.

[31] L. Pouzin, "Virtual circuits vs. datagrams—Technical and political problems," in *Proc. Nat. Computer Conf.*, AFIPS Press, pp. 483-494, 1976.

[32] L. G. Roberts, "International connection of public packet networks," in *Proc. Third Int. Conf. Computer Communications*, Toronto, Canada, pp. 239-245, 1976.

[33] CCITT, "Recommendation X.75—Terminal and transit call control procedures and data transfer system on international circuits between packet-switched data networks," Study Group VII, Temporary Document 132-E, Int. Telecommunications Union, Geneva, Switzerland, Apr. 25, 1978.

[34] D. J. Farber and L. C. Larson, "The structure of a distributed computing system—The communication system," in *Proc. Symp. Computer Communications Networks and Traffic*, Polytechnic Institute of Brooklyn, pp. 21-27, Apr. 1972.

[35] P. Baran, "Broad-band interactive communication services to the home: Part II—Impasse," *IEEE Trans. Communications*, p. 178, Jan. 1975.

[36] R. E. Schantz and R. Thomas, "Operating systems for computer networks," *Computer*, Jan. 1978.

[37] L. Pouzin and H. Zimmermann, "A tutorial on protocols," this issue, pp. 1346-1370.

[38] F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The interface message processor for the ARPA computer network," in *Proc. Spring Joint Computer Conf.*, vol. 36, AFIPS Press, pp. 551-567, 1970.

[39] S. Carr, S. D. Crocker and V. G. Cerf, "Host-Host communication protocol in the ARPA network," in *Proc. Spring Joint Computer Conf.*, vol. 36. Atlantic City, NJ: AFIPS Press, Montvale, NJ, pp. 589-598, 1970.

[40] E. Feinler and J. B. Postel (Eds.), *ARPANET Protocol Handbook.* Network Information Center, SRI International, for the Defense Communication Agency, Jan. 1978.

[41] R. F. Sproull and R. D. Cohen, "High-level protocols," this issue, pp. 1371-1386.

[42] S. D. Crocker, J. F. Heafner, R. M. Metcalfe, and J. B. Postel, "Function-oriented protocols for the ARPA computer network," in *Proc. Spring Joint Computer Conf.*, vol. 40. Atlantic City, NJ: AFIPS Press, Montvale, NJ, pp. 271-279, 1972.

[43] S. M. Ornstein, F. E. Heart, W. R. Crowther, H. K. Rising, S. B. Russel, and A. Michel, "The terminal IMP for the ARPA computer network," in *Proc. Spring Joint Computer Conf.*, vol. 40, Atlantic City, NJ: AFIPS Press, Montvale, NJ, pp. 243-254, 1972.

[44] CCITT, "Recommendation X.21: General purpose interface between data terminal equipment (DTE) and data-circuit terminating equipment (DCE) for synchronous operation on public data networks," *Public Data Networks*, Orange Book, vol. VIII.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 38-56, 1977.

[45] CCITT, "Recommendation X.21-bis: Use on public data networks of data terminal equipments (DTE's) which are designed for interfacing to V-series modems," *Public Data Networks, Orange Book*, vol. viii.2, Sixth Plenary Assembly, Int. Telecommunications Union, Geneva, Switzerland, pp. 38-56, 1977.

[46] ISO, "High level data link control (HDLC)," *DIS 3309.2 and DIS 4335*, Int. Standards Org.

[47] V. Cerf, A. McKenzie, R. Scantlebury, and H. Zimmermann, "Proposal for an international end-to-end protocol," *Computer Communication Review*, ACM Special Interest Group on Data Communication, vol. 6, no. 1, Jan. 1976, pp. 63-89.

[48] A. S. Chandler, "Network independent high level protocols," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 583-602, 1975.

[49] —, "A network independent file transfer protocol," EPSS High Level Protocol Group, 1977.

[50] R. E. Kahn and W. R. Crowther, "Flow control in resource sharing computer networks," *IEEE Trans. Commun.*, vol. COM-20, pp. 539-546, 1972.

[51] L. Pouzin, "Flow control in data networks—Methods and tools," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 467-474, Aug. 1976.

[52] G. V. Bochmann and P. Goyer, "Datagrams as a public packet-switched data transmission service," Universite de Montreal, *Departement D'Informatique Report*, Mar. 1977.

[53] P. L. Higginson, "The problems of linking several networks with a gateway computer," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 453-465, 1975.

[54] J. Shoch, *private communication.*

[55] P. L. Higginson and Z. Z. Fischer, "Experience with the initial EPSS service," in *Proc. Eur. Computing Conf. Communication Networks*, London, England, ONLINE, pp. 581-600, 1978.

[56] D. L. A. Barber, "A European informatics network: Achievements and prospects," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 44-50, 1976.

[57] B. Cross, "General license for message conveying computers," *London Gazette*, pp. 7662-7663, May 28, 1976.

[58] J. Freese, "The Swedish data act," in *Proc. Conf. Transnational Data Regulation*, Brussels, Belgium, ONLINE, pp. 197-208, 1978.

[59] R. Turn, "Implementation of privacy and security requirements in transnational data processing systems," in *Proc. Conf. Transnational Data Regulations*, Brussels, Belgium, ONLINE, pp. 113-132, 1978.

[60] A. R. D. Norman, "Project goldfish," in *Proc. Conf. Transnational Data Regulations*, Brussels, Belgium, ONLINE, pp. 67-94, 1978.

[61] E. Raubold and J. Haenle, "A method of deadlock-free resource allocation and flow control in packet networks," in *Proc. Third Int. Conf. Computer Communication*, Toronto, Canada, pp. 485-487, Aug. 1976.

[62] J. McQuillan, "The evolution of message processing techniques in the ARPA network," *Network Systems and Software*, Infotech State of the Art Report 24, Infotech Information Limited, Nicholson House, Maidenhead, Berkshire, England, 1975.

[63] P. Curran, "Design of a gateway to interconnect the DATAPAC and TRANSPAC packet switching networks," *Computer Communication Networks Group*, E-Report E-67, University of Waterloo, Canada, Sept. 1977 (ISSN 384-5702).

[64] D. D. Clark, K. T. Pogran, and D. P. Reed, "An introduction to local area networks," this issue, pp. 1497-1517.