available from Network Information Ctr., Stanford Res. Inst., Menlo Park, Calif., NIC 4692, 1970.
[8] D. W. Davies, "The control of congestion in packet switching networks," presented at the 2nd ACM Symp. Problems in the Optimization of Data Communications Systems, Palo Alto, Calif., 1971; also this issue, pp. 546–550.
[9] A. N. Haberman, "Prevention of system deadlocks," *Commun. Ass. Comput. Mach.*, July 1969, pp. 373–378.
[10] E. Dijkstra, "Cooperating sequential processes," Math. Dep., Technol. Univ., Eindhoven, the Netherlands, 1965, EWD123; also in F. Genuys, Ed., *Programming Languages*. New York: Academic Press, 1968.
[11] J. Wozencraft and M. Horstein, "Coding for two-way channels," in *Proc. 4th London Symp. Information Theory*, C. Cherry, Ed. Washington, D. C.: Butterworth, 1961, pp. 11–23.
[12] R. A. Scantlebury, "A model for the local area of a data communication network—Objectives and hardware organization," *ACM Symp. Problems in the Optimization of Data Communication Systems*, 1969, pp. 179–201.
[13] K. Bartlett, R. Scantlebury, and P. Wilkinson, "A note on reliable full-duplex transmission over half-duplex links," *Commun. Ass. Comput. Mach.*, May 1969, pp. 260–261.

Robert E. Kahn (M'65) was born in Brooklyn, N. Y., on December 23, 1938. He received the B.E.E. degree from the City College of New York, New York, in 1960 and the M.A. and Ph.D. degrees from Princeton University, Princeton, N. J., in 1962 and 1964, respectively.

During 1960–1962, he was a Member of the Technical Staff of the Bell Telephone Laboratories, Murray Hill, N. J., engaged in traffic and communication studies. From 1964 to 1966 he was a Ford Postdoctoral Fellow and an Assistant Professor of Electrical Engineering at the Massachusetts Institute of Technology, Cambridge, where he worked on communications and information theory. Since 1966, he has been a Senior Scientist at Bolt Beranek and Newman, Inc., Cambridge, Mass., where he has worked on computer communications network design and techniques for distributed computation.

Dr. Kahn is a member of Tau Beta Pi, Sigma Xi, Eta Kappa Nu, the Institute of Mathematical Statistics, and the Mathematical Association of America. He was selected to serve as a Lecturer for the Association for Computing Machinery in 1972.

William R. Crowther was born in Schenectady, N. Y., on June 29, 1936. He received the B.S. degree in physics from the Massachusetts Institute of Technology, Cambridge, in 1958.

From 1958 to 1968, he was a Staff member at the M.I.T. Lincoln Laboratory where he worked on real-time programming of small computers. In particular, he had the full responsibility for the design and implementation of the computer program for the Univac 1218 for the Lincoln Experimental Terminal (LET). He also designed and implemented a scheme for automatic speech recognition and speech compression logic. Since 1968, he has been a Computer Scientist at Bolt Beranek and Newman Inc., Cambridge, Mass., where he has had responsibility for IMP software development and has been involved in the design of other real-time computer systems.

The Control of Congestion in Packet-Switching Networks

DONALD WATTS DAVIES

***Abstract*—Any communication network has a finite traffic capacity and if it is offered traffic beyond the limit it must reject some of it. The data-communication network studied here is one employing packet switching, like the Advanced Research Project Agency (ARPA) network. It handles blocks of data, called packets, and longer messages are subdivided, rather in the same way that store in a computer is allocated in pages.**

A method of controlling congestion is proposed in which there is a finite number of packet carriers in the whole network. When a packet of data is delivered to its destination node the "empty" packet is available for reuse. The empties move randomly round the network and new data must capture an empty packet carrier before being launched into the network. Various elaborations are described that avoid delay in normal conditions. This so-called "isarithmic" method of congestion control supplements and does not replace end-to-end flow control.

Manuscript received May 10, 1971; revised January 10, 1972. This work was performed in part by Plessey Telecommunication Research, Taplow, Bucks., England. This paper was presented at the 2nd Symposium on Problems in the Optimization of Data Communications Systems, Palo Alto, Calif., October 20–22, 1971.

The author is with the National Physical Laboratory, Teddington, Middlesex, England.

Congestion in Data-Communication Networks

ANY communication network has a limit to the traffic it can carry. If there is more than a certain traffic demand, some of the traffic must be rejected. Both the nature of the limitation and the reaction of the network to excess demand depend on the design of the network. The network in a condition where it must reject traffic is called "congested."

The avoidance of congestion is of great importance to public data networks because the facilities they offer will be built in to computer systems that are vital to the operation of trade, industry, transport, etc. Good planning and provision to meet demand is the only means of avoiding congestion. The safety margin needed to guard against congestion will therefore be high for data networks.

In the same way that strenuous efforts are made to avoid faults in computer systems (but the reaction of the system to faults is carefully studied and engineered), so it is necessary to study the reaction of data-communication networks to congestion, even though congestion is to be avoided by adequate provision.

Fortunately the approach to congestion can readily be monitored quite independently of its deleterious effects. Therefore it is not necessary to design systems in which the impairment of performance near congestion is used to give warning of failure. This simple point seems to have been misunderstood by some commentators. We can therefore strive by good design to reduce the effects of overload while planning for the network not to reach this condition and monitoring the safety margin continuously.

This paper concerns the control of congestion in a particular kind of data network that employs packet switching.

Packet Switching in Data Networks

Packet switching is a variant of the message-switching principles used to handle telegraphic messages, but its aims and therefore its design details are different. It is characterized by a low figure of queueing delay, which can be about 10 ms for one transit through a national network using today's technology. A good current example of the packet-switching method is the Advanced Research Projects Agency (ARPA) network [1].

The low delay figure is achieved by employing fast links and short message units. Because the blocks of data to be moved by the network are of variable length but predominantly short, a message unit of 1000 bits or less is proposed. Larger blocks to be moved will be broken into these small units, rather in the way that computer storage is allocated in pages of constant length while the user, unaware of this, deals in segments. The same sort of distinction is made here by calling the customer's unit a "message" and the network unit a "packet." This term is the origin of the name "packet switching."

Communication through the network is usually a matter of sending packets back and forth between two subscribers. While they communicate, a link is said to exist between these subscribers. But a subscriber may also be a multiaccess computer or a cluster of terminals connected to the network through a local packet switch. Such a subscriber must be able to establish many links at one time. It is useful to have a word to describe the individual source or destination of data, whether it is a simple terminal, a process in a multiaccess computer, or one of a cluster of terminals multiplexed by the subscriber. We call it a "socket," using the ARPA terminology. A bidirectional terminal has two sockets, one source, and one destination. Links are therefore established between sockets, bidirectional links employing two sockets at each end.

One purpose of the link concept is to simplify the task of a simple terminal attached to the network. For such a simple terminal, outgoing packets must be assembled and formated by the network and provided with a destination field in their headings. Incoming packets must be rejected unless they come from the link-designated source. Note that the link is a software feature.

A succession of packets moving from source to destination forms a data-communication channel that has a variable data rate. Because links can have variable capacity, a technique of data-rate control must be developed for these networks.

The ultimate limit to traffic in packet-switching networks is expressed in terms of information carried (e.g., packets/second) whereas the limit applying to circuit-switched networks is measured in numbers of calls.

It is possible, in principle, for a packet-switching network to react to congestion by reducing the effective data rate of certain links.

Existing Methods for Control of Congestion

By analogy with road traffic, congestion can be expected to begin at one point in the network and spread as the queues fill and links between switching centres are blocked. Good control of the route taken by packets can increase the load the network will take, but when the limit is eventually reached, several links or nodes will tend to be blocked simultaneously.

Existing congestion-control methods can be classified as local or end to end.

Local control is applied by a switching center on the basis of its own local traffic data (packet rates, queue lengths) also using operational messages received from its immediate neighbors. These messages may request a reduction of traffic over a particular link, or restore unrestricted working, or they may contain data on traffic or delays experienced, etc. They come only from neighbors.

Eventually, as traffic levels increase, the rerouting of packets can no longer prevent congestion, and the network must reject traffic offered to it. If the users who chiefly contribute to an overload are distant from the point of congestion, local control methods require congestion-control measures to spread a long way before effective measures are taken. This is not to say that local control is necessarily ineffective, but it presents difficult design problems.

End-to-end control makes use of the notional links that exist between subscribers (or, more correctly, between sockets). New links that might cause congestion can be refused, in the same way that circuit-switched networks operate.

There are some objections to this method. It controls the wrong parameters because a link is not associated with any particular data rate. The variable-rate nature of links must be preserved, otherwise the designer of a teleprocessing system using the network has to look after data rates as well as the logical structure of the necessary links. A valuable feature of packet switching is the ability to hold open a link economically and thus get

the advantage of rapid response, even though the traffic level may be low or unpredictable. It may require a considerable number of links to serve a big multiaccess system and even if these links are limited to one packet in transit at a time (itself a limitation) this will not prevent congestion. The number of packets it is desirable to have in the system at one time is quite low in order to keep queuing delays small.

Control of the network by a central controller on the basis of all available traffic data could be effective but has bad features: the monitoring and control data increases the traffic load and the controller is a vulnerable part of the system.

Nevertheless, a good control system must react to congestion quickly, and not only locally, therefore some kind of overall control is needed. Such a method is proposed next.

Isarithmic Network

The limitation that may cause congestion, whether at a switching center or over a link, can be expressed to a good approximation in packets per second. Since the average time taken to handle a packet, whether in switching or transmission, is not very dependent on traffic levels, the level of traffic in the network can be expressed quite well by stating the number of packets in transit. This leads us to the idea that congestion can be prevented by placing a limit on the total number of packets in the network.

To achieve this without employing a control center, the number of packets in the network can be held constant. Such a network is called "isarithmic." Since data-carrying packets must be created and destroyed, the balance is kept by using empty packets. Thus when a normal data-carrying packet arrives at its destination it is replaced by an "empty," which is put back into the system. When data are ready to enter the network, an empty packet must be found and replaced by a data-carrying packet.

It is necessary to have a rule for directing empties around the network. The rule should have a random element and in its simplest form it consists of choosing a destination node at random. At the destination, if data are waiting for transit they will use the empty packet, otherwise a new random destination is chosen for them.

The rules for handling empties can be refined as a result of simulation and later by operational experience. The amount of simulation already carried out is small so the proposals given here are mainly based on intuition and need testing. As an aid to intuition, the collection of packets can be regarded as a gas composed of molecules in perpetual random motion. At any switching center or node of the network, packets will arrive and leave at roughly a constant rate whatever the data traffic. Because of the randomness of the motion, all packets will, in time, visit a given node.

The method of operation proposed for the isarithmic network leads to an extra cause of delay, which is the period of waiting for an empty packet before data can be dispatched. This is how access to the network is restricted under congestion conditions, but the designer should try to minimize this delay and hopefully make it very small when the traffic level is small.

Clearly, when a data-carrying packet arrives at its destination and an empty is created, data waiting at that node should have priority for its use. Further than this, it is possible to hold a small store of empties at a node (which is analogous to the gas molecules being absorbed on the surface of the container). The rule now introduced is that a newly created empty goes to the store at once if there is space for it. The size of the stores of empties must be chosen so that a good proportion of packets are left in motion even with no data traffic.

Suppose that the traffic was completely balanced, in the sense that at each node the same number of data packets arrived as departed. There would then be no need for empties ever to be in transit. This condition is unlikely to occur in a real network, but it may happen that the amount of unbalance is roughly known, as a function of time of day, etc. The traffic in empties could then be prearranged, but prearranged flows of empties should not account for more than a certain proportion of them, because the random traffic in empties, which deals with the fluctuations, must not be stopped.

It may prove sufficient for an empty in transit to travel only to an adjacent node and so reduce the journey made before it becomes data carrying again. But this method should be used with caution, particularly in networks of an elongated character, where one end of the network could collect too large a fraction of the available packets. None of the elaborations described is essential to the operation of the isarithmic network, but they are proposed as features that might improve its performance. Simulation is needed to test them.

The basic parameter for an isarithmic network is the total number of packets in the network and to take account of network size this can be divided by the number of nodes to produce a packet-content parameter P. The performance is characterized by C, the total data-carrying capacity in packets per second. This characteristic of C as a function of P depends chiefly on the storage for packets provided at each node, but details of design such as routing, local congestion control, etc., can be used to improve the characteristic. Fig. 1 shows the kind of characteristic expected. For P small the capacity is proportional to P, but as P increases, C reaches a constant value due to network capacity and then decreases as congestion occurs.

The curves in Fig. 1 were plotted from the simulation, in detail, of a particular design of network. The curves obtained are very dependent on the kind of local congestion control applied. In this particular investigation, quite simple changes to the operation of network nodes produced improvements in the characteristics.

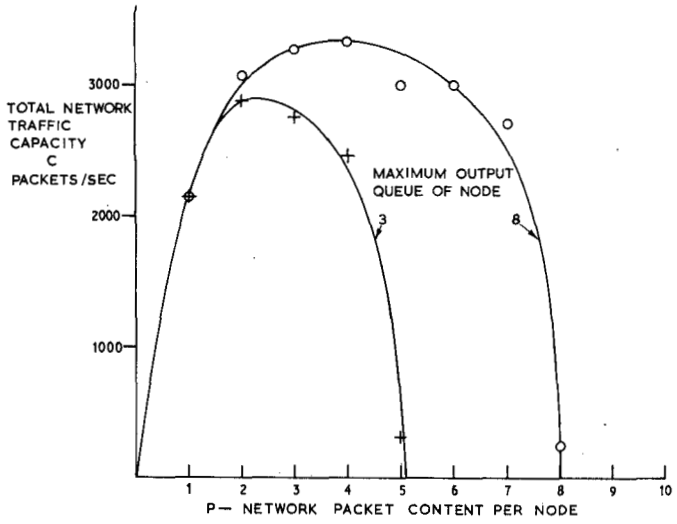

Fig. 1. Characteristic of an isarithmic network (with 18 nodes).

Those plotted here were based on an intermediate stage of development. We did not investigate the precise mechanism of congestion or in particular what caused the complete stoppage of flow at a certain value of P. By better design the saturation can be pushed to higher values of P and the optimum to higher values of C. But the general shape of the characteristic will, it is thought, remain the same. An important parameter in the design was the size of the maximum output queue and curves are plotted for the values 3 and 8. For the former, maximum flow occurs at $P = 2$, for the latter at $P = 4$. The simulated network has on average $3\frac{1}{2}$ output queues per node, so these have on average $\frac{1}{2}$ packet for queue maximum 3 and 1 packet for queue maximum 8 at the optimum setting—a very low value.

It may be objected that control of the packet total does not prevent local congestion by the accidental bunching of packets. This factor showed itself by extreme variability in the simulation for the falling part of the curve, and the points plotted are the worst cases observed. It is hoped, therefore, that Fig. 1 takes into account the accidental bunching of packets.

Any network that is offered traffic beyond its capacity must reject some of it. The isarithmic network rejects traffic at the point of entry to the high-level network, because the rate at which empties become available is insufficient for the demand. In this way the high-level network is protected from seizing up.

The local network is responsible for pushing back to the user the data-rate limitation imposed by this method of control. If no priority rules are used the highest speed subscribers will be the first to be affected. Because of the wide range of data rates employed by subscribers, a limitation on high-speed traffic can preserve the traffic of a large number of low-speed terminals, so there may be some advantages in restricting high-speed subscribers first. Traffic control in a local network is one of the questions being studied in the British National Physical Laboratory (NPL) experimental system [2] where a method has been developed that seems satisfactory in this context.

Earlier in this paper it was asserted that end-to-end control by limiting the number of packets in a link was not sufficient to prevent congestion. When a network is used to connect multiaccess computers to many terminals, the number of links can be very large. The small number of packets in the system at optimum flow, such as the 3–5 per node indicated by Fig. 1, makes any effective control by links extremely restrictive.

Nevertheless, end-to-end flow control over links is needed for the benefit of the users who must control the rate at which they receive data. It is also needed to prevent network congestion because of terminals that have failed to accept packets or to accept them as fast as they arrive. This end-to-end control should be administered as far as possible by the users with network intervention only when the service as a whole is endangered. Isarithmic control supplements and does not replace the flow control over each link.

Operational Questions

Where two isarithmic networks meet, as they might at international boundaries, data may be transferred to new empty packets so that the packet content of each network is preserved. At such a boundary a store of empties rather larger than usual may be created. But the store of data that is required at the boundary might be a source of trouble because it is a packet store within the network that is not controlled by the isarithmic mechanism. As long as international data traffic is a small part of the total traffic the problem can be dealt with by giving international traffic some priority, where it enters a country, in the use of empties.

According to the details of the design adopted, there are various parameters such as the packet content, routing of empties, and size of store for empties that can be varied to optimize the network operation. These parameters can be changed during operation and they can be made to vary throughout the day to take account of different traffic conditions. Such additional elaborations would only be adopted after a good understanding of network behavior had been gained.

The isarithmic network shares some of the vulnerability of central control, because a node fault could result in a steady gain or loss of packets. To give a rough idea of time scale, assume that a node can process a packet every 5 ms and there are 100 packets in the network. If a node is losing all the packets it receives, the half-life of the packet content would be roughly $\frac{1}{2}$ s. But if positive acknowledgment is used, it seems unlikely that this kind of fault can happen. If it does prove a problem, an independent monitor of packet input and output could be added to each node.

The packet content can be altered and packets loaded

initially from any node. To make a census of packets it suffices to have a single bit in each packet to record whether it already has been counted. Packets can be counted as they arrive at one particular node. As long as there is some traffic at each node, stored empties will not be retained, but it might be necessary to generate data traffic in exceptionally quiet conditions in order to release the stored empties for counting. The routine messages that ask for traffic data from each node will do this.

To summarize, isarithmic operation is one method for the prevention of congestion inside the network. The method can be elaborated in very many ways, the most significant being the provision of stores for empties. Simulation study is needed to discover which of the many elaborations are worthwhile. An extra feature that may need to be added is a method of monitoring packet gains or losses in fault conditions.

References

[1] L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," in *1970 Spring Joint Computer Conf., AFIPS Conf. Proc.*, vol. 36. Montvale, N. J.: AFIPS Press, 1970, p. 543.
[2] R. A. Scantlebury, "A model for the local area of a data communication network—Objectives and hardware organisation," presented at the 1969 ACM Symp., Pine Mountain, Ga.

Donald Watts Davies received degrees in physics and mathematics from the Imperial College, London, England, in 1943 and 1947, respectively.

He took part in the early development of stored-program computers, helping to build the ACE pilot model from an initial design by A. M. Turing, which first ran programs in 1950. He is now Superintendent, Computer Science Division, British National Physical Laboratory, Teddington, Middlesex, England.

A Communications Interface for Computer Networks

DONALD KARP AND SALOMON SEROUSSI

***Abstract*—The scope of this paper is to describe the architecture of a communications line protocol for computer networks. Development and implementation details will be introduced where necessary to clarify the presentation.**

The need for an architecture to facilitate interprocessor communications has been a requirement to the computing industry for several years. The described line protocol was derived through an experiment with a computer network designed for heterogeneous machines, and which utilized existing software. Due to the inflexibility encountered by this approach, the architecture is being reimplemented using our own software.

The line interface was defined with flexibility as the foremost requirement. The protocol developed utilizes a minimum set of line-control characters. Information is passed in the header portion of the transmitted block providing the capability of identifying a wider range of line-control and user-related functions. Error recovery has been implemented based on the same type of messages and by transferring line timing responsibilities from the hardware to the software.

Manuscript received November 30, 1971. Part of this paper was presented at the 2nd Symposium on Problems in the Optimization of Data Communications Systems, Palo Alto, Calif., October 20-22, 1971.

The authors are with the Computer Sciences Department, IBM, the Thomas J. Watson Research Center, Yorktown Heights, N. Y. 10598.

I. Introduction

THE DEVELOPMENT of computer networks has generated a need for an efficient computer-to-computer line protocol that will handle interactive communications between heterogeneous computing machinery. The requirements exacted by a computer network are such that presently available communications protocols can only satisfy them by means of cumbersome, time consuming, and highly inflexible procedures. The scope of this paper is to describe an interface that will satisfy the requirements for this type of network. It is our intention to show a simple network communications access method (NCAM), with line-control procedures based upon considerations essential to the design of a flexible system, and structured to make optimum use of all available resources.

This experimental version of NCAM is being implemented to run under control of IBM operating system 360-370. It should be noted, however, that the protocol is designed to operate in a half-duplex or full-duplex environment and with other types of hardware and software configurations.