THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 51, No. 6, July-August, 1972
Printed in U.S.A.

Network for Block Switching of Data

By J. R. PIERCE

(Manuscript received December 4, 1970)

This paper describes a possible sort of data network for the transmission of addressed blocks of data between data terminals. The network consists of a number of closed rings or loops around which data blocks circulate. Data terminals on local loops can read data into empty blocks and read data addressed to them from full blocks. Buffered devices transfer data between local and regional loops and between regional loops and a national loop. Loops so interconnected need not be synchronized; they need not even operate at the same bit rate. Provisions are made for a "busy" signal, for special loops to carry heavy traffic between particular local loops, and for alternate routing.

I. GENERAL INTRODUCTION

1.1 *The Aim of This Paper*

This paper describes a data network in which addressed blocks of data find their way through a network of interconnected loops. The network does not use common control. Rather, simple customer terminals and simple devices which monitor and interconnect "loops" or "rings" would be added as such a network was brought into being. Thus, the cost and complexity would grow as the network grew.

1.2 *The Digital Communication Situation*

There is an increasing amount of digital traffic associated with computers and computerlike devices. Some of this traffic is highly intermittent but demands good transmission and quick response. Most networks at present available either offer quick response but monopolize a channel (private lines) or require an appreciable setup time (line switching).

One approach to handling intermittent, quick-response traffic is to transmit such traffic in blocks around loops or rings.[1–6] This paper describes a system in which simple customer terminals put blocks of data onto or take blocks of data off of a loop or ring. Nonsynchronous loops or rings can be interconnected to form a widespread network.[6] This network takes digital circuits as they are; it does not impose special modulation formats as has been proposed.[5]

II. THE ENVIRONMENT OF A SWITCHED DATA NETWORK

The environment in which a switched data network would operate is different from the environment of telephone service. About all one can say for sure about "data" is that the input and output will appear in, or can be translated by the user into, binary form, and into serial binary form if need be. It is commonplace for the user or his machine to address messages, to look for addresses, to check for errors in various ways, and to perform other complex functions. Thus, user equipment can do, the writer believes, many things that the carrier might consider doing.

A data network would be different in another way. The line-switched telephone network is already there, in reach of wherever a data terminal or a data switching equipment might be. Hence, it is not at all necessary that messages concerning faults or the monitoring of operations be transmitted by the data network itself. The line-switched telephone network can be called in at any time.

Finally, in order to succeed, a data network must be able to grow gradually, gracefully, and economically, both geographically and in traffic capacity. Data channels for growth can readily be provided. Switching means will be needed at an increasing number of locations as the service grows. Switching means, which must of course be highly reliable, should be as simple and inexpensive as possible, consistent with providing satisfactory service to a community of sophisticated users.

III. SERVICE REQUIREMENTS

What service requirements should we put on an address-switched data communication network? What should be reserved for the carrier and what should be left to the user?

In addressing these questions we should remember that equipment or functions "left to" the user can as an option be supplied by the carrier through special "add on" equipment.

In the list below, several areas of requirements are discussed.

3.1 *Modulation and Timing*

The modulation and timing should be under control of the carrier;

otherwise there can be no assurance that transmission facilities will work or can be interconnected. The user should be provided with binary digits and a timing wave, which can be used in accepting digits or in putting digits into a terminal.

This argues strongly the desirability of performing all network functions other than transmission on binary streams, and providing modems to convert to binary at the ends of all transmission circuits which are used.

3.2 *Signal Format*

It is assumed that messages are transmitted by means of addressed blocks of binary digits, blocks of a common length. Certain positions within a block must be reserved for synchronizing and supervisory functions. Other positions should be accessible to the user for message digits and addresses. The sender's address might be put on by the carrier (to avoid misuse) or by the user. There seems no reason why the user should not supply the destination address.

3.3 *"Privacy"*

The carrier should supply in its switching system or data terminal means to assure that a message reaches only the customer to which it is addressed.

3.4 *Blocking*

A certain probability of blocking is inevitable.[7,8] However, provision should be made to prevent a single user from blocking a system for a prolonged time by transmitting continuously. This could be achieved by assuring that the user puts data into the system at an average rate considerably lower than the speed of the channel which serves him.[9] Thus, he would necessarily leave the channel idle after transmission of any block.

3.5 *Errors*

Nothing that the carrier can do can entirely prevent errors. The cost of error correction can be high. Different users may tolerate different error rates. The computer art is sophisticated. It seems best to leave error correction to the user. The carrier should endeavor to supply a low-error-rate service. If this could not be done over certain transmission links without error correction, the carrier might use error correction in these transmission links. The carrier can of course offer error correcting equipment as an option.

3.6 *Failure to Deliver*

Some messages are bound to fail to reach their destination, either because of blocking in transmission, errors in transmission, or because the user is out of operation or busy. The simplest way to deal with failure to deliver is acknowledgment of receipt. This can be left up to the user. If the user divides a long message into many blocks, he can, if he wishes, number the blocks and acknowledge only the first block, the end block, and loss of an intermediate block (when he receives a block whose number is not one greater than that of the preceding block which he received.) Such precautions could take care of the possible but unlikely time interchange of blocks—a block being delayed by alternate routing and arriving after a block which was sent later.

3.7 *User Busy*

In different systems, "user busy" can mean different things. It might merely mean that the user wasn't reading things out of a buffer fast enough. This would certainly happen if user equipment failed. The carrier could provide some notification to the sender if a terminal fails to accept a message intended for it.

3.8 *Failure of Service*

The system should somehow monitor its performance and request service when it fails. Alternate routing can allow service around a failed portion of the system. When the user believes that service has failed, the most sensible provision would seem to be a telephone number that would put him in touch with a computer or a person.

3.9 *Buffering at Data Terminals*

Some buffering will probably be necessary at data terminals to allow adequate notification of receipt of data and to allow notification of time when a block can be transmitted and correct timing of transmitted data. Buffering beyond this minimum required amount could be provided by the user, with an optional offering by the carrier.

IV. A PARTICULAR NETWORK

This section describes a particular data network. It makes use of closed digital loops, as in some other networks.[1–5] However, taken as a whole, it differs from other loop networks.[1–5] It is assumed that access to transmission facilities is through modems whose inputs and outputs are binary digits and a timing wave.

Figure 1 shows several interconnected loops which could form a small data network or a part of a large data network. Each loop is a data transmission channel. T1 lines look very attractive for local loops to which customers are connected. Other lines, such as 50-kilobit lines, could serve as trunk loops.

Three sorts of boxes appear in the loops:

Each loop has a box A which contains a clock and a buffer, so as to time and close the loop. A also performs other functions.

Unless a loop is a trunk loop, it also has boxes B which put blocks of data on and take blocks of data from the loop.

Loops are interconnected by boxes C, which transfer blocks of data from one loop to another and perform other functions.

Loops need not be synchronous, and speeds of transmission on different loops can be very different. Thus, transmission means can be fitted to the traffic on the loop. In transferring blocks from one loop to another, buffering will be provided to take care of differences in bit rate. Commonly, the buffer size will be one or more block lengths.

The operation of the network in Fig. 1 is best explained by considering particular features and blocks.

4.1 *Block Format*

Figure 2 shows a possible block format. The block is divided into several sections.

One section provides bits used for synchronization. These are written in by an A and used by B's and C's.

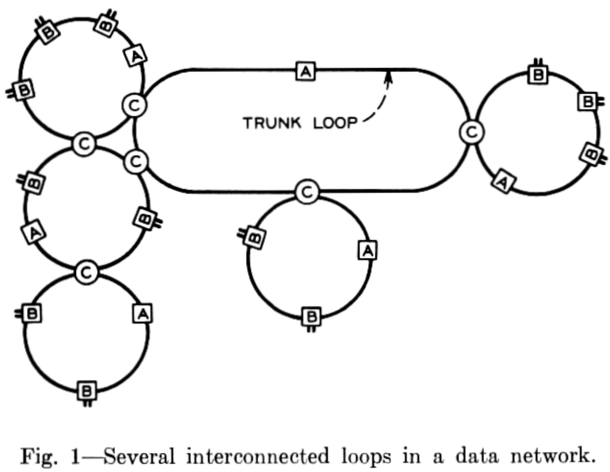

Fig. 1—Several interconnected loops in a data network.

Another section indicates whether the block is vacant or full; this reads vacant until a B or a C writes in *full* prior to writing address and message digits into the block.

Another part of the block is reserved for other supervisory marks; this space is used by the A's, B's, and C's, as will be explained later.

Other spaces in the block are reserved for destination address, sender's address, and message digits. These digits are accessible to the user on receipt of a message and, with the possible exception of sender's address, are supplied by the user.

4.2 *Box A*

An A contains a clock and circuitry for writing in the synch digits at the beginning of an outgoing block. In addition, it contains a buffer for storing the digits of an incoming block. The buffer can conveniently be a whole block long.

A block written by a B may have to pass an A in order to reach another B on the same loop. In this case, the A will simply read the bits of the block out from the buffer into the corresponding positions in the outgoing block. However, an A *marks* any full block that passes through it. When a marked block passes through an A, the A can interchange the sender's and destination addresses of any marked block. This returns the block to the sender and so provides a "busy signal" (the returned block) in case of any failure to deliver.

When the addresses are interchanged, the block should be marked so as not to be erased if it passes an A box once but to be erased if it passes *any* A box twice (rather than another interchange of address). This keeps undeliverable blocks from circulating endlessly through the network.

Rather than interchanging addresses in a marked block, an A could simply erase a marked block. In this case the system provides no "busy signal."

The A box should monitor the incoming signal and somehow send

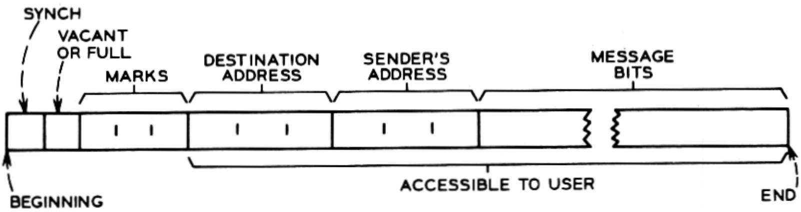

Fig. 2—Block format.

an alarm signal if there is none, or in the case of any other detectable malfunction.

4.3 *Box B*

The purpose of box B is to write data from a source into empty blocks and to read data from blocks.

B has to synchronize an internal counter or clock with the synchronizing pulses A writes in the blocks. This can be done by various methods.

In putting a message into a vacant block, B will have stored in its buffer part or all of the bits to go into the block. B will monitor the loop for a vacant block. When it finds one, it will (*i*) mark the block as full, (*ii*) signal its message source that it is transmitting so that more bits can be read in from the computer or other source, and (*iii*) write in address and message bits. B may write in the sender's address from the message source or from an internal store.

In taking a message from a block, B will examine the destination address of any full block. If this is the address of that B, the B will (*i*) signal the computer to accept bits, (*ii*) read address and message bits into buffer (if desired, message bits only can be outputted to the computer), and (*iii*) mark the block as empty and erase any harmful supervisory marks.

If desired, box B can signal the computer whenever a complete block has been received or transmitted, as well as when reception or transmission starts.

4.4 *Box C*

The purpose of box C is to transfer blocks from one loop to another. As the loops which box C interconnects may have different bit rates, buffering up to several block lengths may be necessary or desirable.

A C must decide whether to transfer a block or not. Thus, it must examine a part of the destination address of the message. The part it must examine is the address *of* the loop on which the destination lies, not the address *in* the loop on which the destination lies. The examination should certainly not involve a lengthy table lookup.

In Section 4.6 the writer describes one simple scheme. In this, the C need merely compare a part of the destination address with one particular address and determine whether the addresses are the same or different.

When a C transfers a block from one loop to another, it marks the block on the first loop as empty and makes any necessary erasures.

When a C effects a transfer, the block it puts onto the new loop is marked as not having passed an A; otherwise the block would be erased in passing an A in the new loop. The C, however, leaves intact a supervisory mark which shows that destination and sender's addresses have been interchanged.

An appropriate system layout will allow alternate routing; a message may be transferred by any of several C's on a loop, but not by others.

If the buffer of a C is full when a message which it might have transferred arrives, the C rejects the message. A message rejected by all C's on a loop could pass an A twice, and the A would either interchange sender's and destination addresses, so that the sender would get a "busy signal" (the returned block), or else erase the block. However, if the addresses had already been changed and the block so marked, the block would be erased on passing a particular A for the second time.

A C with a full buffer should signal for help. If the buffer is often full, there is danger of blocking. If the buffer stays full, something is wrong.

4.5 *Lost Blocks and Other Troubles*

Several things can cause the loss of a block.

Mutilation of address (through errors in transmission or in box function) can render a block undeliverable. Such a block will eventually be erased or returned.

Buffer overflow in a C may result in rejection of a block and subsequent destruction at an A. This is the *carrier's* fault; more buffering or more channel capacity could have prevented loss.

A block may reach the proper B but be rejected because of inadequate buffer capacity. If this bothers the recipient, he should add more buffer capacity.

A block may reach the proper B but the equipment for which it is intended may be down.

Errors introduced in going through the system may alter the address and send a block to the wrong destination. If the number of possible addresses is much larger than the number of addresses in use, this is not very likely to happen. Users can guard against false receipt of blocks by reserving some of their message bits for identification of desired messages, or by accepting messages only if the sender's address is in some restricted group.

4.6 *Systems Plan and Logic*

In the foregoing account the exact logic of the C's and the overall organization of the system have not been spelled out.

Figure 3 indicates the general features of the system plan. There are several hierarchies of loops—three in the figure. These have been labeled L (local), R (regional), and N (national). There is only one N loop. Various R loops connect to it by C's. L loops connect to the R loops by C's. L loops have B boxes on them to serve customers.

An address of a message destination (or a customer) consists of three parts:[10,11]

n_1 , Number *on* the local loop (L)

n_2 , Number *of* the local loop (L)

n_3 , Number of the regional loop (R).

Let us consider how a C should act in transferring blocks between loops:

(*i*) A block on an L loop is transferred to an R loop if its destination n_2 (number of the local loop) is *different from* the number of the L loop it is on.

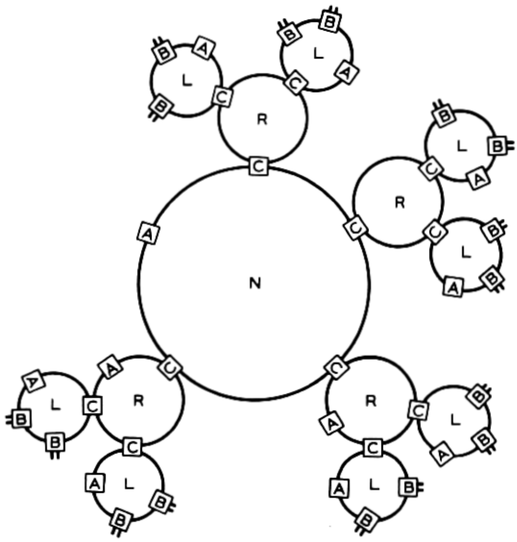

Fig. 3—General features of system plan.

(*ii*) A block on an R loop is transferred to the N loop if its destination n_3 is *different from* the number of the R loop it is on.

(*iii*) A block on the N loop is transferred to an R loop if its destination n_3 is *the same as* the n_3 of the R loop to which it can be transferred.

(*iv*) A block on an R loop is transferred to an L loop if its destination n_2 is *the same as* the n_2 of the L loop to which it can be transferred.

It is easy to see that this logic always gets a block to the destination address.

The above scheme does not provide alternate routing, and it channels all traffic between different regional loops or their local loops through the national loop. There may be heavy traffic between particular pairs of regional or local loops. To accommodate this, R or L loops can be interconnected by special TRUNK loops, as shown in Fig. 4. The C's for such interconnection should be located so that a block passes them just before it reaches the C to a higher-order (N or R) loop, as shown in the figure. The C's to the TRUNK transfer a block only if the destination n_3 (for a TRUNK connecting R loops) or the destination n_2 and n_3 (for a TRUNK connecting L loops) is the same as the n_3 or n_2 and n_3 of the loop to which the TRUNK can take the block.

Sometimes one might wish to connect a physically remote R or L loop to an N or R loop. This might be done by a physical extension of the remote loop, as shown at the left of Fig. 5. It might be more economical to provide a trunk loop, as at the right of Fig. 5. Thus, blocks on

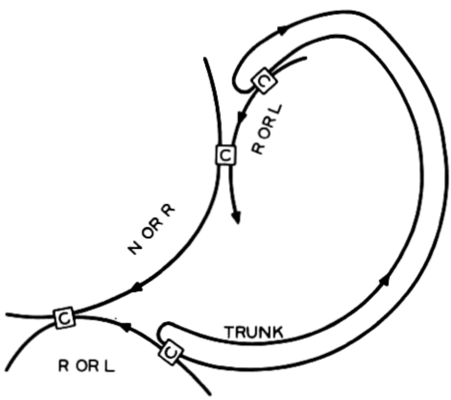

Fig. 4—Special TRUNK loop.

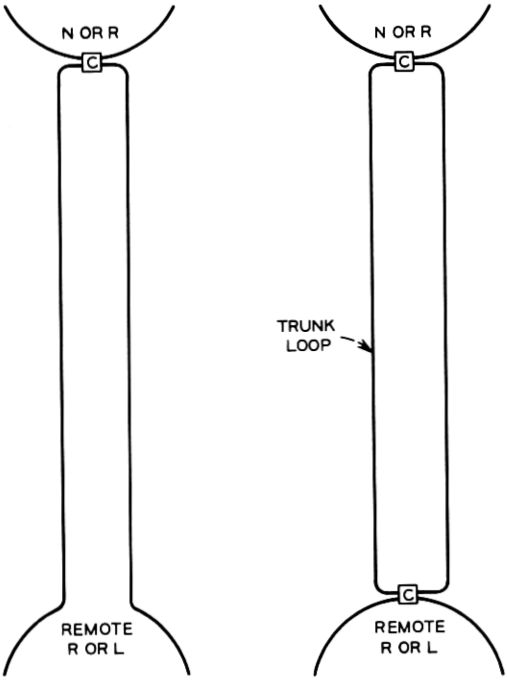

Fig. 5—Methods of connecting to a remote loop.

the remote loop would not have to travel the length of a long extension. The trunk loop might be a slower-speed loop than the remote loop. The two C's should have the same logic as the one C to the left.

What if the N loop fails? We can provide alternate routing by interconnecting regional loops with ALTERNATE loops as shown in Fig. 6. The C's connecting the R loops to the ALTERNATE loop should come *after* a C connecting an R loop to the N loop. Ordinarily, a block whose destination n_3 is not the same as the n_3 of the R loop it is on will be transferred to the N loop. If this fails, the following C will transfer the block to another R loop, as shown in the figure. Further ALTERNATE loops can be arranged so that a block which fails to reach the N loop will be passed from one R loop to another until it finally reaches the right R loop.

Reliability can also be increased by making the N loop and all R loops double and interconnecting them as shown in Fig. 7.

The L loops can be made double, too, perhaps by using the common two-way equipping of T1 lines. In the case of the L loops, one of the

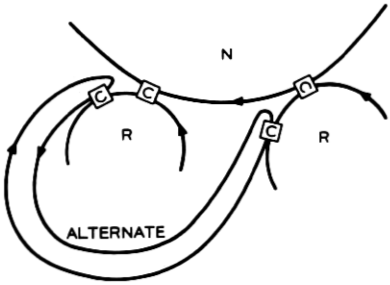

Fig. 6—ALTERNATE loop.

loops should be used in preference until it fails; then the A, the B's, and the C's should automatically shift to the other loop of the pair.

If a B loses power or fails, an automatic switch should bridge across it so as to keep the loop closed.

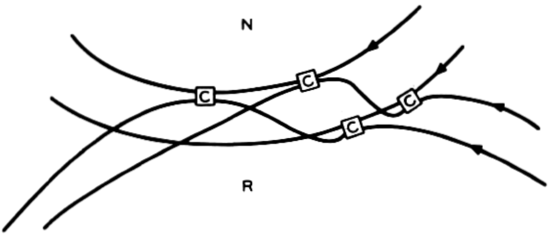

Fig. 7—Another method of increasing reliability.

4.7 *Blocking and Traffic Considerations*

The fact that material to be transmitted may be held for the appearance of a vacant block means that it is somewhat more likely that a full block will be followed by a full block than by an empty block. This must be taken into account in analyzing blocking.

It has already been mentioned that an increase in buffer size at a C should decrease blocking.

V. SOME CONCLUDING THOUGHTS

Unless the network is overloaded, the data network outlined in the preceding sections will get a block of data from source to destination very quickly.

Transmission of blocks of fixed length may seem restrictive, but it has a number of advantages. It allows easy synchronization at B's and C's and sure location of positions within the block. It lends itself to acknowledgment and retransmission. Combined with a requirement that the average rate at which a user can transmit be considerably lower than the loop rate, it prevents one user from blocking a loop.

If the C's are to have a simple criterion for transferring blocks, the numbering of loops must satisfy some simple criterion. The freedom available through translation between directory and office numbers is not available. Translation would certainly run the cost up and might slow operation. Perhaps it is wise to forego translation in a data network, and stick to a carefully planned numbering scheme.

The data network has been deliberately kept very simple. It is multiprocessing with a vengeance. A few standard modules do everything. The logic and much of the circuitry of all A's is the same; adaptation is to the speed of the line. The same can be said for the B's and C's. Of course, for different types of channel, different modems will be needed in going to and from binary and in extracting and using the timing wave. Nonetheless, the network could be constructed of A, B, and C boxes and existing circuits and the modems for them.

REFERENCES

1. Company brochures on the Collins C-8500 Direct Digital Control System.
2. Grosser, H. K. M., and Schramel, F. J., "Data Transmission and Switching Equipment for the Seat Reservation System of United Air Lines (1)," Philips Telecommunication Review, *24*, Feb. 1963, pp. 13–24.
3. Martin, James, *Teleprocessing Network Organization*, Englewood Cliffs, N. J.: Prentice Hall, Inc., 1970, pp. 122–125.
4. Steward, Edgar H., "A Loop Transmission System," Proceedings of the 1970 International Conference on Communications, San Francisco, June 8, 9, 10, Institute of Electrical and Electronics Engineers, Inc. (70-CP-370-COM), pp. (36-1)–(36-9).
5. Farmer, W. D., and Newhall, E. E., "An Experimental Distributed Switching System to Handle High-Speed Aperiodic Computer Traffic," *Proceedings of ACM Symposium on Problems on the Optimization of Data Communication Systems*, Oct. 13–16, 1969.
6. Pierce, J. R., Coker, C. H., and Kropfl, W. J., "An Experiment in Addressed Block Data Transmission Around a Loop," IEEE International Convention Record, March 1971, pp. 222–223.
7. Hayes, J. F., and Sherman, D. N., "Traffic Analysis of a Ring Switched Data Transmission System," B.S.T.J., *50*, No. 9 (November 1971)., pp. 2947–2978.
8. Avi-Itzhak, B., "Heavy Traffic Characteristics of a Circular Data Network," B.S.T.J., *50*, No. 8 (October 1971), pp. 2521–2549.
9. Another solution is given in Kropfl, W. J., "An Experimental Data Block Switching System," B.S.T.J., this issue, pp. 1147–1165.
10. Kropfl proposes a slightly different way of handling n_2 and n_3 in Ref. 9 above.
11. Graham, R. L., and Pollak, H. O., in "On the Addressing Problem for Loop Switching," B.S.T.J., *50*, No. 8 (October 1971), pp. 2495–2519, propose a quite different addressing scheme.