An APL Users Meeting

Toronto September 18,19,20,1978

Sponsored by:
I. P. Sharp Associates Limited

A DATA COMMUNICATIONS SYSTEM FOR THE APL USER

Roger D. Moore
I.P. Sharp Associates Limited
Toronto, Ontario

Introduction

SHARP APL began operation in the summer of 1969 with a network which contained one time division multiplexor and sixty dial ports. By early 1976 this had grown into a medium-sized collection of time division multiplexors including one trans-Atlantic link. During 1976 and 1977 this network was replaced by a packet-switched network. The primary goal of the network, both before and after the change, has been the provision of APL service to remote low-speed terminals. When introducing new technology it is desirable to remedy the drawbacks of the old without sacrificing its benefits. This general goal resulted in some specific, attainable objectives. It also revealed areas where compromise was required. This paper attempts to record various observations and decisions made while building a data communication system for the APL user.

User Interface

The network should not be particularly obvious to the user. Initial connection is kept as simple as possible. With a single input, the user can identify both terminal type and destination APL machine. If the requested APL facility is available, user interaction with the network ends. After the connection has been established, the network attempts to act as a simple forwarding agent: output from APL is passed to the terminal for printing, and input from the keyboard is passed to APL. Unlike certain public networks there is no provision for escape to the network during a session. The closest approach to this is the network recognition of XON/XOFF when the terminal is not in the "input to APL" mode. These control characters allow the terminal to control the rate at which output is sent from APL to the terminal.

The network requires a certain amount of information about the user's terminal. This information is required for terminal control and for printing legible error messages. The APL\360 trick of examining the initial right parenthesis to resolve BCD vs. Correspondence has been retained. This character was chosen on the grounds that many people would accidentally type it during a sign-on attempt and the information required by the network would be acquired. The initial right parenthesis from an ASCII terminal is not actually used by the network. It is retained briefly so that an obfuscative blot can be elicited from APL after a connection has been established. It is also used as the left character of the destination selection command. The ASCII character set issue is largely avoided by the absence of special characters in messages from the network.

TDM Drawbacks

Prior to the installation of the present packet-switched network, I.P. Sharp Associates (IPSA) used older technology to accomplish a similar function. Some of the deficiencies of the time division multiplexor (TDM) technology hastened the conversion to a packet-switched technology.

Telephone lines are a known source of transmission errors. This is particularly true when efficient modems are used. An efficient modem provides a transmission rate (measured in bits per second) which exceeds twice the bandwidth (measured in hertz) used. Although this gives a rather efficient use of the analog voice channel, a small number of errors (perhaps one hundred per hour) are introduced. With the TDM technology of several years ago, most of these random errors tended to be visible to the APL user. Erroneous data would be printed on the terminal or sent to APL. Some method of protecting the user from minor errors on the network trunks was required. The usual method of compensating for short error bursts is to group several characters into a block and to append some redundant bits to the block. The receiver can then perform some arithmetic operation upon the received block including the redundant bits to see if a certain class of error has occurred. If the arithmetic operation in the receiver indicates an error, then the receiver requests retransmission of the block. The Sharp network uses the same redundancy check as does HDLC (x*16+x*l2+x*5+l residue). This gives the ability to detect all error bursts of less than seventeen bits and most bursts of greater length.

The network does not provide any protection on the lowspeed link between the terminal and the network. In some locations this continues to be a source of error. Control of this error source requires some sort of retry protocol between the network and the terminal. Little work has been done in this area by IPSA.

Simple TDM technology dedicates one channel to every port passing over a particular trunk. The channel assignment is made at installation time. It cannot be altered without briefly removing the multiplexor from service (or in one extreme case returning it to the factory). This fixed assignment of bandwidth can impose limitations on the total number of ports in areas where digital bandwidth is not free. When the entire terminal population consisted of 134.5 bps terminals of the same character length, this was not a particularly onerous restriction. It was possible to squeeze about fifteen 134.5 bps channels into a 2400 bps highspeed trunk. The same 2400 bps can only accommodate eight 300 bps channels.

During the period of transition from 134.5 bps nine-unit terminals to the present 300 bps ten-unit terminals, autospeed presented a further complication. It is desirable to have a single telephone number for both kinds of terminals. With simple TDM technology it was necessary to have separate telephone numbers and separate channel groups for the two terminal classes. By using a technology without fixed bandwidth allocations the transition from 134.5 bps BCD to 300 bps ASCII was eased.

Packet-switching has allowed a bit of gambling on bandwidth requirements. The total dial-up capacity at a remote site can be arbitrarily large. There is no "a priori" bandwidth allocation for a particular port. Bandwidth is allocated on demand exactly as CPU time and file bandwidth are allocated within APL. If an overload exists along a particular route, the symptom is a general slowdown of output. This is quite noticeable to the APL user as printing will occur in bursts corresponding to individual packets. This may be caused by a general overload situation such as when trans-Atlantic traffic is running over a single trunk rather than the normal two. An alternative cause is a transient overload due to a trunk going out of service for a second or two. If this happens during a peak period a transmit queue can build up which will take several seconds to clear down to an innocuous level. Even under light load conditions, a brief service interruption on a trunk will exhaust the endnode buffer pool for a printing terminal. The resulting delay will be noticed by the user.

The location of the pause (mid-line rather than inter-line) is a side effect of the method of packing APL output lines into network buffers. The educated user can use the location of the printing pause to discriminate between poor performance in the APL machine and network degradation. The observed network degradation may be caused by either an acute or chronic overload. Output is held at the APL machine (or in network transmit queues). This was considered preferable to loss or corruption of output, and it has also influenced pricing. A 120 cps terminal has a lower priority on transmission from APL than does a conventional terminal. This may reduce the mean print speed observed at the 120 cps terminal to less than 120 cps. For this reason, imposition of a connect time surcharge on 120 cps terminals seemed rather unfair.

If a fixed bandwidth had to be assigned to every dial-up port, it would be difficult to accommodate the number of ports demanded by the typical optimistic branch manager. With dynamic contention for bandwidth, a relatively large number of ports can be installed at a site to accommodate peak connect loads. There is a gamble that not all ports will be continuously engaged in printing. A terminal which is not being used for output requires zero bandwidth.

Packet-switching Drawbacks

After concluding that simple TDM technology was unacceptable because of the problems mentioned above, some choices still had to be made in a packet-switched environment. Although TDM technology does have certain drawbacks it also has certain user benefits. If all of the allocated ports are being used to print full speed it is quite efficient. It can almost always provide very good response time.
Most data transmission systems require some extra bits which are not directly useful to the end user. For example, the asynchronous terminals used with most time-sharing systems have start, parity and stop bits associated with each character. None of these is directly useful to the end user, but they are required by the asynchronous protocol. Similarly, a TDM requires some extra bits to maintain framing synchronization. A packet-switched system is even worse. It has a redundancy check for each block. Each block also contains addressing and sequencing information to allow it to be delivered to the proper port in sequence. With the present Sharp network packet size, about fifty percent of the bits transmitted are overhead bits of no direct value to the user. With a TDM, the overhead content is closer to twenty percent. The dynamic bandwidth allocation capability of the packet system tends to balance the higher efficiency of the TDM system. Thus, the number of 30 cps terminals which can be comfortably accommodated in the packet system is only slightly higher than the number for a TDM system (twelve versus eight assuming 2400 bps trunk).

Packet-system efficiency is theoretically easy to alter. The number of overhead bits per packet is fixed for a particular protocol. By increasing the number of useful bits in a packet while preserving a constant overhead, the apparent bandwidth efficiency (useful bits / total bits) can be increased. Unfortunately, the need for acceptable response time constrains packet size.

Delivery of a packet from APL to an endnode may require forwarding the packet down more than one link. The time delay imposed by one link (`T[I]`) is given by the formula:

```
      T[I] ← CD[I] + LENGTH÷BPS[I]
```

`CD[I]` is a constant delay associated with the link and its modem. A typical value for an intracontinental link is between 10 and 20 msec. `BPS[I]` is the link speed in bits per second. `LENGTH` is the packet length in bits. If the subscript vector `PATH` selects the links required to reach a specific endnode, the delay to the endnode is `+/T[PATH]` or

```
  (+/CD[PATH]) + LENGTH ]× +/÷BPS[PATH]
```

The network currently uses a 256-bit maximum length packet. For a 2400bps link, the length sensitive term in the delay is 106 msec. Thus the observed delay with one or more 2400 bps links in the path is rather sensitive to packet length. Doubling the packet length would change bandwidth efficiency from the present .5 to around .75 but it would also increase delay by one or two hundred milliseconds in certain parts of the network.

A possible solution to the capacity versus response tradeoff is to vary the packet size with the terminal speed and location. A more complex solution would be to use short packets for the first two packets of an APL response and then revert to a larger packet length.

One important design consideration in a packet-switched system is the most likely form of use. For an APL-oriented network, the expected form is point-to-point communication with relatively long holding times. A single port with a connected physical terminal is associated with a particular terminal task (T-task). Any interaction with other ports in the network will be via APL facilities for intertask communication, rather than through network facilities. Not all computer networks have this point-to-point property. In particular, a process control system which monitors many different sensors and sends commands to many devices does not have this point-to-point property.

A network with an orientation toward point-to-point communication is often referred to as a virtual call network. It resembles a switched telephone system in which physical circuits are established and held to accommodate a particular point-to-point call. Certain administrative packets are exchanged to set up a virtual call; this precedes the transmission of any data. In the Sharp network this occurs after the initial input line (normally "o)") and before the security blot is printed. After the call has been set up, an abbreviated form of addressing can be used. This is possible because there is need to identify the originator of a packet within a virtual call. There are only two ends to a virtual call; the end which receives a packet assumes it came from the other end.

Addressing and Routing Considerations

An alternative form of organization is the datagram network. In a pure datagram network every p includes the name of the packet originator as well as the destination. In theory this allows any to conduct simultaneous dialogues with all other ports in the network. For some applications this can be useful, but time-sharing does not seem to be one of those. Datagrams are used within the Sharp network for certain administrative purposes such as call setup and status monitoring. It appears these uses could be replaced with virtual calls with only moderate inconvenience. Converting the Sharp network to a strict virtual call organization would provide certain simplifications. In particular amount of control information required to route datagrams is proportional to the square of the number of nodes in the network. In a pure virtual call network, routing information need only be computed and maintained for existing rather than potential virtual calls. This reduces the quantity of routing information to the number of virtual calls times the mean path length. As the size of the net is currently around sixty nodes and continually increasing, datagram addressing is becoming rather
unwieldy. Changing to a pure virtual call network appears necessary for further growth.

Another advantage of a fixed-route virtual call system is that one of the more unpleasant theoretical properties of a store and forward network can be eliminated. Most retransmission and queuing schemes do not provide an upper bound on packet delivery time. It is also possible in some systems to lose a packet without notification being provided to the sender. The usual approach to the lost administrative datagram problem is for the sender to wait a finite time for a response. If no response is received within the timelimit, the sender assumes (perhaps incorrectly) that the original packet or the response has been lost.

With virtual call addressing, a unique call number is to assigned to every virtual call in the system. By constraining all traffic to a point-to-point format over fixed routes, two problems can be solved. The packets for a particular virtual call will be stored in the forwarding queues of a small identifiable set of nodes. This is not true for elaborate dynamic routing systems. When the call is completed or aborted, the packets of the call can be removed from the network. For a call which is being completed in an orderly fashion, the precaution of procrastinating the call completion signal until the data pack have been forwarded is adequate. When a node or a link along the route of a virtual call fails, packets are assumed lost and the virtual call must be aborted. An abort signal is passed from both sides the failure to the endpoints of the virtual call. As the abort signal passes along the route it has the side effect of destroying any packets belonging to the virtual call. The virtual call number is then available for reuse, as the previous generation of packets with the same virtual call number has been destroyed.
One other drawback to the existing scheme of routing by node number is that traffic passing between a pair of nodes reachable by independent paths is constrained to a single path. One reason for the constraint is the simple route table organization. Within each node the route table is stored as a vector indexed by destination node. Each element of the vector gives the direction in which some other node lies. As the current route table entries are scalars, it is impossible to encode an alternate route. A more serious problem is the mechanism which aborts virtual calls after a link failure. It is extreme important that both endpoints of an aborted virtual call receive notification of the abnormal call termination. When this mutual termination requirement is combined with a requirement that calls not using the failed link remain undisturbed, the termination and routing algorithms impose a condition upon the route tables. The route from A to B can be expressed as an ordered list of intermediate nodes. With the present routing algorithms, a necessary and sufficient condition for mutual call termination after link failure is that the route from node A to node B must be exactly the reverse of the B to A route for all node pairs. When every virtual call has its own route, the consistency requirement only applies to individual virtual calls. The route from end A to end B must be the reverse of the B to A route for that virtual call. A simple forwarding validation algorithm can guarantee this in realtime. Other virtual calls with identical endpoints can be routed independently.

Link Management

One area of concern in the initial design was maintaining performance in the presence of minor line errors. A trans-Atlantic link running at 7200 bps can store four or five unacknowledged packets. Some link management protocols such as X.25 require retransmission of the entire pipeline after a single error. This has the advantage of preserving packet order over a single link. It has the drawback of exaggerating the effect of a single error. Instead of retransmitting the single corrupted packet, four or five packets may be retransmitted. The Sharp network retransmits only the corrupted packet. This has the drawback of permuting the packet order along the link. If the various packets in the pipeline were associated with independent virtual calls, this would be of no consequence. The order within a virtual call could be disturbed by retransmission. In a network which uses multiple routing, differing transmission delays along distinct paths can give the same effect. The possible ill effects of order loss are avoided by including an interend sequence number in every packet belonging to a virtual call. The receiving endpoint can then process the packets in the order in which they were emitted rather than the order in which they are received.

APL Interface

The network would be of little value without a connection to the APL machine. Several methods of connection were discussed during network design. An obvious solution would be to preserve symmetric hardware of a TDM system. At the computer room end, data from a single virtual call

would materialize on a physical port attached to the network. This physical port would be electrically connected to a Memorex 1270 terminal controller. This solution is sometimes called lowspeed demultiplexing. It is simple in concept and appears exactly like an ordinary 1270 configuration to APL. In addition to the obvious problem of requiring large quantities of hardware, lowspeed demultiplexing has other drawbacks. The worst is the way in which the entire system is slaved to a 300hz oscillator in the terminal controller. Data transfer to and from APL is constrained to a single procrustean speed. There are situations where transfer at a rate which is either faster or slower than the normal terminal speed is desirable.
Running the traffic into the APL machine on a single highspeed line and demultiplexing within APL machine was contemplated. This was rejected mainly because of the excessive changes which would have been required in the APL software.

The solution which is now in use involves placing a stored program computer between the APL machine and the network. This is a true frontend computer as it is attached directly to a 360 byte multiplexor channel. On the channel side, it looks very much like a MRX 1270 or IBM 2703. Every virtual call uses a separate address on the byte multiplexor channel. The frontend computer trivially adjusts the rate at which data is transferred from APL to the network. This adjustable rate is fairly important in achieving acceptable response time. When APL starts generating output in response to an input, all of the network buffers for that virtual call are empty. In this situation, data is transferred at high speed over the byte multiplexor channel to fill an initial buffer ration. Data transfer for that virtual call then stops until a buffer has been sent to the terminal. Another buffer is then filled with data from APL and sent through the network. This process occurs outside of APL and is not usually visible. The network buffers are independent of APL buffer size and APL output line boundaries. If the network can absorb output faster than APL is generating it, a partial buffer will be sent. A typical case is the *SENT* response to *)MSG*. When APL is generating output more rapidly than the network is willing to accept it (⎕←⍳1*E*4), the normal APL mechanism for suspending an output-limited task is used.

Flow control is also useful on input to APL. SHARP APL has retained the 1969 APL\360 scheme of demand assignment of input buffers. The default assumption in APL is that an input line will be quite short. If an input line is longer than expected, extra buffers will be assigned within APL. It is theoretically possible for the network to present data faster than APL can allocate buffers. To avoid loss of keyboard input, a method of regulating the rate at which the frontend computer presents data to APL was required. This was accomplished by sending an explicit end-of-buffer signal from APL to the frontend computer whenever an APL buffer is filled. The frontend computer suspends data transfer for the affected virtual call until the APL machine signals that a new input buffer ready. This signalling is normally effected by command chaining on the byte multiplexor channel and thus takes less than sixty microseconds. If intervention by APL software is required to allocate a new buffer, the "buffer ready" signal may be delayed for an arbitrary period. The data is held in the frontend computer until the "buffer ready" signal is received from APL. This solution avoided the potential embarrassment of carefully forwarding (and perhaps retransmitting) a packet over the ocean and then corrupting it with a race condition in the computer room.

Establishment of a new call is another area where a frontend computer was useful. The number of ports in the network somewhat exceeds the number of T-tasks which can be simultaneously supported with the present IPSA hardware configuration. This makes dynamic allocation of byte multiplexor channel addresses desirable. When the frontend computer signals a new connection to APL, control information pertaining to the call is transmitted to APL. The terminal protocol (ASCII vs. BCD), terminal speed, port location and the user's right parenthesis are passed to APL. The APL software needs all of the information except port location to communicate with the user's terminal. The port location is not directly used by the APL system, but is available for various applications (particularly operational statistics) which need to know the geographical location of a terminal.

The diagram below illustrates the topology of the network. Each rectangle represents a node in the network and each line represents a link. About sixty of the nodes shown are currently operational. The diagram includes some planned installations and some nodes used for hardware and software testing. The only redundant link in the network is in the 1 4 43 32 loop in the lower right-hand region.

The program which draws the diagram normally uses an APL terminal as an output device. This places some constraints on the diagram (particularly on line direction). The version shown here was drawn by an HP7221 plotter with assistance from Bob Bernecky.
TOPOLOGY DIAGRAM REMOVED/RDM

Conclusion

Conversion from TDM technology to packet-switched technology has been done with minimal change to APL. Both the internal structure and the external appearance are approximately as they were before the introduction of packet-switching. There has been some sacrifice of response time in order to achieve greater reliability and capacity. The greatest present and future benefit is the ability to provide reliable service in many different geographical locations.