INTERCOMPUTER COMMUNICATIONS IN REAL TIME CONTROL SYSTEMS

S. R. Smith, R. W. Goodwin, and M. R. Storm
National Accelerator Laboratory*
Batavia, Illinois

Summary

Considerations of bandwidth, noise, distance, response time, data and system integrity, maintenance, interfacing, software control, data organization, cost, and development time complicate the implementation of intercomputer links in real time control and data acquisition systems. This paper tells how to do it. The NAL link provides an example.

Introduction

This paper provides guidelines for design of intercomputer links for real time control and data acquisition systems.

Definitions

An "intercomputer link" allows an application or user program in one computer to communicate with a similar program in another computer. A link therefore consists of a transmission channel between computers, the hardware interfaces between the channel and the computers, and the software routines in each computer necessary to run the hardware. Information transmitted over a link is divided between data and control information. Data is the answer to: "Why have a link?". Control makes the link work. Control information may include error checking, acknowledges, interrupts, block length and identification.

"Real time systems" are those systems which collect and process data quickly enough to provide outputs useful for affecting a process as it occurs. This definition is similar to the ones used by Yourdon[1] and Martin[2]. Required response times for real time systems typically vary between a few milliseconds and a few seconds.

Philosophy

Providing a good solution to engineering problems requires that the solution set be limited. Solving too general a problem requires excessive development time, will probably cost a lot of money, and may result in a system too powerful for the actual application. Complicated systems often prove cumbersome for simple applications and are more difficult to comprehend and maintain than simpler systems. "Entities must not be multiplied without necessity".[3] If new hardware and software must be designed, this is particularly important. Everyone underestimates development time.

The system designer must also avoid underdesigning. A system that collapses when requirements increase slightly is a worse solution than one too powerful for the job at hand.

The designer must first identify the bounds imposed by the system on the link. The design of a link within those boundaries must allow for probable changes. Real time systems are notorious for expanding beyond initial requirements.[1,2] Required data rates can easily double, for example.

Most real time computer link problems can be well defined. The computers involved are specified. Distances are fixed. Data rates have known upper bounds. Response time requirements are known. This circumstance allows the design of links powerful enough for any likely requirements yet not outrageously expensive or complex.

Topology

System topology is the first parameter defining a link. Topology includes the number of computers (nodes) and the connections between computers (branches). The connections may have direction for either data or control. Most of the data transmissions may be in one direction. Perhaps one computer is a "master" so control flows from master to slave.

Changes in topology can improve system performance or reduce the cost. For example, suppose several computers in one location all require a fast link to a computer in another location, but each remote computer requires use of its link only in short, infrequent bursts. One solution is to have a separate link from each remote computer. A better solution might be to provide local multiplexing among the remote computers so they could share a single fast link.

Channel Selection: Geography and Bandwidth

Geography puts physical dimensions on topology: How far apart are the computers? Bandwidth is another dimension: How much data goes where? How fast? Computers may be across a room from each other or across a continent. Data rates range from teletype speed to the cycle time of integrated circuit memories.

Average data rate is not a useful parameter for specifying link performance. Data transmissions occur during a small part of the total response time of a real time system so that the balance of the time can be used for processing the data and acting on it. Rates in different directions may be different, as in data acquisition systems with little control capability. Peak data rate and system geography combine to define the data transmission channel.

The Channels

Figure 1 summarizes the applicability of various communication channels as a function of link length and bandwidth. The least expensive hardware channel is shown where several channels overlap.

---

* Operated by Universities Research Association Inc., under contract with the U. S. Atomic Energy Commission.

Applications below 10^4 bits/second are dominated by character oriented bit serial (e. g. teletype) channels using or designed to use voicegrade telephone lines. For short distances, modems and phone lines are unnecessary. RS232C or similar signaling methods on dedicated wire pairs provide an inexpensive channel for distances from 10-1000 meters depending on cable quality and speed. Common carrier voicegrade lines support asynchronous communication up to 2000 bits/second, or synchronous communication up to 10,000 bits/second, for unlimited distances. Computer manufacturers commonly provide interfacing hardware and software for phone line channels. Control information uses special characters.[4,6]

Computers separated by less than 100 meters may be linked by a dedicated multipair cable. A word can be sent in parallel from one computer to another. Bandwidth is limited to about 10^7 bits per second. Separate pairs in the cable carry control information, simplifying the interface problem.

High quality dedicated coaxial cables can support serial bit streams at 10^5-10^7 bits/second over hundreds of meters. If the bandwidth of the cable is much larger than the bit rate, simple pulse coding may be used. If the data rate approaches or exceeds the cable bandwidth, more sophisticated techniques such as phase modulation are required. It is cheaper to buy good cable and use a simple code.

Information bits and timing belong on the same cable. Codes which provide timing along with information bits are so simple to encode and decode that there is no reason to use separate cables. Since the serial transmission format is under the control of the design engineer - rather than being fixed, as in a "teletype" phone link - control information as well as data can easily be multiplexed onto the same cable. This is done be preceding each transmitted word by several bits identifying the remainder as control or data.

Two cables - one for each direction - containing data, control and timing information support full duplex transmission with simple transmitter and receiver logic. Use of more than two cables is a waste of copper.

Bit rates up to 10^7 bits/second can be achieved using two coaxial cables. This is one computer word every 1-4 μsec, more than sufficient bandwidth for all but extreme applications. Word rates this high tax the I/O capability of most computers.

There are few cases where data rates faster than one computer word every several microseconds are really necessary. If such a rate is maintained for even a few milliseconds, an enormous volume of data will accumulate. Even the most powerful computers cannot respond intelligently to thousands of words of data in less than many milliseconds. Gathering data that cannot be processed is pointless.

In extreme cases, an order of magnitude speed improvement over bit serial can be obtained by transmitting several bits of a computer word simultaneously on separate dedicated coaxial cables. Usually some fraction of a computer word such as a 6, 8, or 12 bit byte is sent in parallel. Separate signals can identify the byte as control or data.

Shared memory is a simple, very fast intercomputer link. If the memory is random access for both computers, merely changing a location used for control "transfers" an entire block.

Common carriers provide private, wideband lines with speed up to 2×10^6 bits/second over hundreds of kilometers. At the higher data rates, these are expensive, but alternatives are hard to come by.

In some cases, dedicated radio links can be used instead of wideband phone lines. If one end of a channel is moving, radio links may be the only possibility.

Real time systems requiring data rates in excess of a couple of million bits per second over more than a thousand meters need to be redesigned. Perhaps the computers can be moved closer? Do you really need that much data that fast? If there is no way out of this problem, you have got a development project on your hands. Any system with a requirement like this will be a major project in any case, so the link problem will be small by comparison.

Channel Errors

All channels require some form of error detection and/or correction to insure that mutilated data is not accepted as being good. Real time systems often require better protection against errors than other systems. Acting on bad data in a real time environment can have immediate and disasterous results. If errors are detected, there may not be enough time to allow retransmission within realtime response requirements. Real time systems do not get a second chance.

Secure channels with bit error rates smaller than 10^{-9} can use simple parity checking on each word. This case includes short runs of low speed pairs without modems, high speed serial systems using coaxial cables, parallel multipair channels, and shared memory. Good grounding and shielding practice should be followed on dedicated cable systems of any kind. When the long distance signal standards are under the control of the designer, high output levels and high receiver thresholds insure low error rates. Noise on long runs of dedicated coax can easily be one volt peak to peak measured differentially on terminated cable.

Channels that involve a common carrier line experience bit error rates between 10^{-8} and 10^{-2}.[4,5] Radio links in particular are subject to high error rates and fading. Avoid radio links if possible.

The data communication industry has developed many sophisticated techniques for dealing with errors in high error rate situations. Methods such as automatic equalization, geometric coding, polynominal coding, forward error correction, data scrambling, and retransmission schemes are adequately treated elsewhere, and will not be discussed here. Martin[4] gives a good introduction to this subject.

Use as secure a channel as you can afford, it will save many headaches.

How to Select a Channel

To select a channel, locate your application somewhere on Figure 1. If the point falls near the center of a region, that is the channel to use as far as hardware considerations go. If the point falls near a boundary, look at the methods for adjoining areas. Remember that rate requirements easily increase, so the next faster method for a particular distance is a good bet. Prudence dictates use of as fast and as secure a link as is reasonable.

Of course, a fast channel can be used for a slow application. Multipair cables or shared memory offer simplicity of operation compared to teletype-like channels even if the required data rate is low. Likewise, long distance techniques work for short links. Serial channels using two coaxial cables work over 10 meters.

Software is a major and always underestimated expense in any real time system. Cost of providing software to operate a link often exceeds the link hardware cost.

Most computer manufacturers supply hardware and software for character oriented communication over voicegrade phone lines. Many manufacturers offer fast parallel links or shared memory. Some offer interfaces to wideband phone links.

If manufacturer supplied equipment suits your application at all, use it! It will be suited to the computer. It will save development pains and cost. It will probably work.

Usually standard hardware can be used. Real time systems can sometimes use standard software. If so, most problems such as error control, system integrity, maintenance, reliability, interfacing details and perhaps link control protocol have been solved.

Applications exist that are just not covered by existing equipment. Perhaps the speed/distance combination requirement indicates a channel not available for a given computer. Perhaps several brands of computer are involved. High speed ports for different computers are almost always incompatible. Perhaps the manufacturer's software just does not do the job for the real time system. OK, its a do-it-yourself project.

The Channel/Computer Interface and Link Software

The interface and link software connect the channel and the application program in the computer. The choice of a computer and a channel are usually independent. The channel is chosen on geography and bandwidth. The computer may be chosen on computer power considerations, because the boss likes it, because it is new, because software is available for it, because it is already owned, or because the manufacturer is low bidder. Computers that adapt well to real time systems have modern I/O facilities such as interrupt driven I/O and independent, asynchronous, or buffered I/O processors. These features simplify life for the interface engineer and the programmers who design the link routines.

An application program should be able to specify to the operating system only the destination (or source) and characteristics of a data block. The link software makes sure the transmission occurs properly and reports to the application program the completion of the operation, or the failure if the situation is hopeless. Application programs should not be concerned about details of the channel, the interface or the link protocol.

A good interface relieves the link software of burdens like acknowledge interlocks, separation of control and data messages, error detection, and as much of the protocol as possible. Complicated link software is evidence of poor interface design.

Buffering and Acknowledges

Except for some very low data rate applications, the computer on the receiving end of a link will not generally be ready to accept a word arriving from the channel when it arrives. The interface must provide temporary storage in a receiving register. If the receiving register still holds the first word when another word arrives, the second word either will disappear or will clobber the first word. There are two solutions to this problem: an acknowledge interlock or a larger buffer at the receiver.

An acknowledge interlock prevents the transmitter from sending another word until the previous word has been accepted by the receiving computer. The receiver sends an acknowledge to the transmitter when the receiver holding register is emptied, indicating readiness for another word. If propagation time is not excessive or if overlapping is used, this method is a good, simple solution. A "negative acknowledge" sent back in case of an error can cause retransmission.

A larger buffer at the receiver allows the receiving computer to fall behind by the size of the buffer. In any case the speed of the receiving computer, averaged over the buffer size, must be at least that of the transmitter, or data will be lost eventually.

Links often use a hybrid of these methods. The unit of transmission is a block of words less than the size of the receiver buffer. Positive or negative acknowledgment occurs at the end of each block.[4]

Buffering at the receiver allows the receiving computer to input data while more data is arriving. Similarly, a one word buffer at the transmitter allows the transmitting computer to output a word while the previous word is being transmitted. Thus, single word buffers at both ends allow computer I/O time to be overlapped with the transmission time. This technique substantially improves total link speed when channel transmission time is close to computer I/O time.

Another use for large buffers arises when the transmitting computer must unload its data quickly to tend some other job. The computers may not be able to communicate simultaneously. The buffer holds the message until it can be transmitted.

Link Protocol

Before computers can communicate meaningfully, they must agree on the direction, size, and the meaning of a data block.

If one computer can be a "master", the other a "slave", protocol can be simple. The master always controls the link. The master initiates a transmission by sending a couple of control words to the slave. This control information specifies the direction, size, and contents of the following data block. Receipt of the control word may cause an interrupt at the slave computer to gain its attention. The master then sets up its I/O port to transmit or receive as appropriate. Meanwhile, the slave sets up to transmit or receive the specified block. When both computers are set to go, the interface hardware starts the transfer which continues until the end of the block. Data acquisition systems commonly use this technique.

Networks organized more democratically require more complicated protocol. Computer A interrupts computer B, accompanying with a control word or two. This is a request for transmission. A waits. When B is ready, B answers A's request by granting (or denying) a transmission. Both set up their interfaces and transfer the block.

These procedures vary if, for example, provision must be made for interrupting a long block for a crucial message. Perhaps B answers A's request by simply setting up its channel. If the channel is not set up, no data moves, and A will know this by inspecting the block count, timing out if there is no change.

Full duplex communication will require additional I/O ports and is not worth the effort unless really required.

It is an advantage to be able to send control information along with a request. Otherwise, many short transmissions must be made to transfer control information. This means that there are two logical data paths involved, one for control and one for data. Hardware should supply this facility if possible even though another channel may be required. The control channel needs only a fraction of the bandwidth of the data channel.

Maintenance and Reliability

Most real time systems are required to operate continuously for days or even weeks. In the event of failure, operation must be restored quickly. The problem of maintenance must be faced early in the design cycle if it is to be solved at all. The maintenance problem is a strong argument for a simple modular design, particularly if operating personnel perform maintenance outside of normal working hours.

When part of a large system fails, the first and most complicated problem is to find the offending component. This problem is reduced if the rest of the system continues to operate and assists in identifying the broken module. Link software should specify mode of failure: Which link is down? Which end is down? Is the error rate too high? Is the other computer responding at all? Diagnostic programs that (try to) exercise specific pieces of hardware are invaluable. Diagnostic lights and test points liberally sprinkled thru the hardware expedite identification of problems.

Operating personnel should be able to localize the failure to a fairly large piece of hardware. An entire satellite computer could be replaced with a spare!

Once the cause of a failure has been corrected, the whole system should recover automatically.

Yourdon[1] treats real time system failures and recovery in detail.

The NAL Intercomputer Link

The NAL control system provides an example of intercomputer link design for realtime systems.

The topology for the NAL computer control system is a central computer linked to several satellite minicomputers. The central computer is the master and exercises control over the links to the satellite computers. Since data originates in the accelerator, the bulk of the information flows towards the central computer. The volume of information required for control is small by comparison.[7]

The satellite computers at NAL are located between 20 and 1000 meters from the central computer. Peak data rates vary between 10^5 bits/second and 5×10^5 bits/second. When these points are plotted on Figure 1, it is clear that NAL should use a channel consisting of two dedicated coaxial cables.

NAL has developed a channel with bandwidth of 10^6 bits/second over 1200 meters of good coaxial cable. The transmission scheme uses ± 12 volt signals on terminated 50 ohm cables with pulse transformer coupling on both ends. Receiver threshold is 2.4 volts. Error rate is much lower than 10^{-9}. The low error rate justifies use of simple parity checking for error detection. Bit serial transmission rate is 2.5 MHz. Data transmissions contain 24 bits including 16 data bits, start, stop, control, and parity bits. Hardware generates an acknowledge interlock for each word. Control transmissions contain 6 bits including 3 control bits, start, stop and parity bits. The 8 transmission types encoded by the control bits include data, acknowledge, negative acknowledge, and system control signals. The hardware supports full duplex transmission at a rate of one word every 13 μsec in each direction, not counting propagation time. The central computer port is half duplex, so the full duplex hardware capability is not used.

Transmitter and receiver registers are buffered to allow the I/O time of both computers to be overlapped completely with the serial transmission time.

The NAL link uses I/O processor ports on both the central computer and the minicomputers. The minicomputers CPU's run at 2/3 full speed during link transfers. Each active link requires 1/3 of the central computer I/O capability.

The central computer starts transmissions by initializing channel hardware at both ends, then interrupting the minicomputer and sending it a control word. The control word specifies the direction, contents and length of the following data block. The block must start within 600 μsec and words must be transferred every 400 μsec after that or the central computer aborts the transmission. Errors also abort a data block. Central computer overhead for each block is 750 μsec. Up to 32 different types of transmissions are possible in each direction to any minicomputer. Maximum block length is 1023 sixteen bit words.

If a minicomputer has a message for the central computer, a flag is set in a group of words read by the central computer at the 15 Hz NAL injector rate. The central computer responds by fetching the message from the minicomputer in a standard transmission.

Failure in a minicomputer does not affect the rest of the system. Failure of the central computer does not affect the minicomputers. If a link is broken, the system recovers automatically upon repair of the link. Operators can "cold load" any minicomputer from the central control room.

References

1. Edward Yourdon, Design of On-Line Computer Systems, Prentice-Hall, 1972.

2. James Martin, Design of Real Time Computer Systems, Prentice-Hall, 1967.

3. William of Occham, 14 Century English Philosopher.

4. James Martin, Teleprocessing Network Organization, Prentice-Hall, 1970.

5. Bell System, Pub. 41007, 1971.

6. Computer, February 1973, several articles.

7. R. E. Daniels, NAL Computer Control System, Paper H1, 1973 Particle Accelerator Conference.

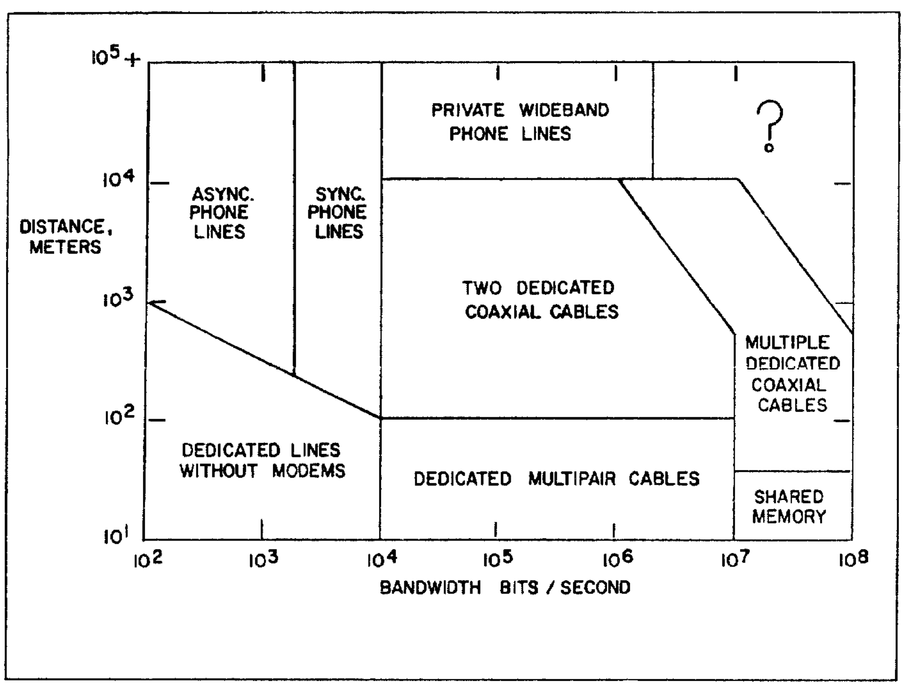

Fig. 1. Channel hardware as a function of distance and bandwidth.