THE FABLE OF "REAL-TIME"TELEMETRY DATA MOVEMENT

Thomas W. Gustin, Senior Design Engineer
SYSTRAN Corp., Dayton, Ohio

ABSTRACT

This paper presents an exciting new concept in real-time information distribution that can be easily integrated into existing and future telemetry reception and data dispersal systems. After briefly examining the evolutionary path and various perceptions of the concept "real-time", a variety of techniques are explored in achieving the expedient movement of real-time information. Many non-telemetry application environments are now using real-time shared-memory networking techniques to obtain large, high-speed integrated sharing of common information. The phenomenal results are partially attributable to high reliability, extremely low latency, and ease of use. This paper attempts to present various telemetry applications and scenarios with descriptions of benefits achieved by simply changing existing data movement techniques to those using shared-memory networking techniques. KEY WORDS: fiber optics, GOLD RING™, local area network, real-time, SCRAMNet™.

INTRODUCTION

The evolving computer technology and the unquenchable widespread "need" for information have forced a comparable, yet divergent, growth in information dispersal techniques. Time-based throughput, specifically from when information exists until when it is available for use, has been one common understanding of the "real-time" concept, and the evolving data dissemination methodologies have enabled this subset of "real-time" to diminish to a micro-portion of that which it used to be. Depending upon certain conditions, some currently used message-passing protocols may soon become extinct due to the growing demands upon the information sharing techniques. A few of the more recent versions already have vaulted into new areas of capabilities that are not attainable by some of the older ones. Much of this evolution has been spawned by ever-changing "real-time" requirements.

Early telemetry systems helped to define some of the older computer system techniques in data movement. It appears that now it is the computer systems' turn to boost telemetry data movement from quasi-inert methodologies into a realm of "real-time" and "in-the-loop" system performance capabilities; a dimension thought unreachable until just recently. It is this evolutionary path in data movement as it can be applied to telemetry systems that is the focus of the following discussion.

"REAL-TIME" EXPOSE'

Years ago an early implicit understanding of "real-time"computer throughput involved getting my program printout back within an hour, instead of overnight. A more recent telemetry-related comparison of data availability involved receiving the bound computer printout of PCM-collected event data within one week of the test, instead of one month. Granted, both of these data movement examples lack "real" urgency, but they adequately demonstrate typical throughput capabilities for their system resources and intended uses. If my telemetry test results were available the same day as the test, experiment parameters and the phenomenon monitored could have been modified for an immediate repeat test on the following day. These represent actual experiences for many experimenters, especially during unconstrained tests. Therefore, "real-time" may be defined as information transit time; from data existence to availability for use.

McGraw-Hill's Dictionary of Electronics and Computer Technology remarks about real-time include: data processing systems that control an ongoing process and deliver results no later than the time when they are needed for effective control; or a computer system that operates in real time; or the handling of information at a rate sufficient to ensure that it will influence the operation under control at the required time. These real-time definitions express that real-time is rather loosely defined.

Often a computer's worst-case response time is defined as a measure of its ability to operate in real-time. Since there are so many different ways of measuring such a parameter (interrupt latency, context switching time, etc.) it is difficult to place much credence in numbers without standards for measurement that are generic enough to be applicable for all machines, to avoid comparing apples and oranges. Satisfaction of the real-time need ultimately depends on the application, and many machines or no machines may be able to satisfy such timing requirements.

The terminology "real-time" is often made synonymous with "frame-time" in both telemetry and simulation environments. Especially for the simulation environment, the frame time, and hence the real-time period, is usually chosen to be short enough to ensure the subjective sense of realistic operations. For telemetry test applications, the frame time is chosen to maintain a sufficiently high enough sampling rate so that when the data is "played back", it will provide a realistic motion picture of the actual events that were recorded through the telemetry system. For "in-the-loop" telemetry control systems, the frame-time definition of real-time closely approximates the dictionary's presentation for process control applications.

One may legitimately ask why is real-time operation so important? In process control applications not running in real-time, the catastrophic results would be evident. It is far more subtle when aliasing errors are introduced during data acquisition operations due to a frame time that was too long. An aircraft simulator not operating in "real-time" may manifest this problem by jerky visual displays or sluggish responses to control manipulations. In many cases, what looks fine to one person is totally unacceptable to another, depending upon experience and other technical factors.

Trying to locate or derive a more precise definition of real-time is a little like trying to catch a spark, it may be done but it may not be necessary. There are now effective ways to eliminate the bottlenecks of speed in achieving "real-time" performance in many types of systems, and these will be discussed shortly. While it is obvious that the quantitative measure of speed is a very important element in the concept of real-time, it is not so evident that there are also subtle qualitative figures of merit also.

The next task is to identify several options available for the movement of information from source to destination within the revised operational envelope of "real-time" operations.

INFORMATION DISPERSAL

There are many different methods of transferring information from its source to its destination(s). The reasons for selecting one method over another are as varied as those for any other system configuration decisions; including (but not limited to) cost of material and labor, user friendliness, best solution available, reliability, fit into long-range plans, factory support, retrofit minimum impact, broad acceptance by others, etc.

A few of the slower (non real-time) information movement techniques have been presented already. For stand-alone systems, those involving only one backplane where all involved resources have fair accesses to information, real-time information movement is not a problem. Real-time communications has been a problem since the trend of using multiple, linked-together subsystems became a viable alternative to performing all operations within one supercomputer/system. As the speeds and performance capabilities of these subsystems increased with technological improvements, fast and reliable intrasystem information sharing techniques have grown comparatively worse due to their inability to keep pace with other system improvements. Over chronological time, many different types of subsystem components with varying capacities have been combined into useful quasi-supercomputer complexes with an arsenal of memory, user interface and communication techniques.

For computational systems based upon a single vendor or architectural concept, linking multiple resources together to perform a supercomputer's task load for real-time operations is typically accomplished using a physical shared-memory scheme. The obvious advantages of minimum information transit time, low or no software overhead, fast error recovery, and quick system reconfiguration potential are very appealing to the system architect. However, when another computer from a different architecture needs to be added to accomplish a slightly larger task, or some of the machines need to be spread out or moved beyond the physical link's capabilities, or the access and protocol schemes are really only efficient for small numbers of on-line users, the disadvantages become overwhelming.

The most common solution to the disadvantages of a physical shared-memory technique of moving information between sources and users is the usage of one or more forms of a message-passing network. The advantages of message-passing networks over physical shared-memory methods of sharing information is that great distances can exist between larger numbers of resources, resources that can be based upon completely different architectural approaches. Figure 1-1 presents a partial tree of some of the more common methodologies used to perform information movement, based upon, their network/media access schemes. Note that point-to-point communications is not represented explicitly, although many of the communication classes presented can be used for just two resources sharing information. It is also important to point out that there are other ways to represent networks besides their access methods.

Within the envelope of the revised "real-time" definition most of these protocols can be eliminated as serious contenders for generic real-time communications due to one or more of the following: very low throughput performance values because of inefficient physical links; the need for software drivers to perform some or all of the communications and interface functions, robbing valuable CPU time; poor or non-existent error recovery schemes; message length performance dependency; and nondeterministic operations due to protocol methodologies, arbitration schemes, and message collisions for random uncontrolled accesses to the media.

An ideal "real-time" network would have information transit times approaching those of normal memory interactions, completely deterministic behavior, total transparency (no software overhead), host platform independency, large node number capacity, long-haul capability, proven reliability and industry confidence. It would have all of the combined advantages of physical shared-memory and message-passing networks, and none of the disadvantages. At the system level, all sharing resources would have no contention (immediate) access to the shared data pool, and would do so by merely reading from and writing to memory as needed by the application program; no I/O driver subroutine calls. Due to the time-associative nature of “real-time” applications, the data needs to be moved from the source to all of its destinations (for usage) on time, every time; no exceptions for collisions or media access arbitration allowed. Higher reliability can be achieved by using fiber-optic media to eliminate EMI/RFI error susceptibility, and novel forward error detection/correction techniques for error free information passing. These characteristics must hold true regardless of message length, one byte to millions of bytes.

If performance generalizations were made to include all operations involved with information movement, then the list would include resource contention/access times, collision recovery times, error detection and correction times, CPU time to process and handle information, and actual information movement times. Lumping together the times for these activities, where applicable, and then using each method to move medium to large amounts of information to be shared between resources, a comparative diagram will result as depicted in Figure 2-1. Recall that the working definition for “data movement transit time” is that time span from when the information first exists until it is available to the user. For most information movement techniques, there is far too much “overhead” time required to perform the actual information movement. Arguments and usage stipulations supporting each type of networking technique, where throughput would be maximized, can be proposed; but in general, the relationships hold as depicted for medium to heavy network bandwidth utilizations. For instance, many of the data movement techniques suffer from indeterminant performance characteristics due to access methodologies that are seriously degraded when more than one shared information resource is “broadcasting” data. Many of the other methods were designed to avoid access collisions by using a more controlled access scheme which actually hampers potential information throughput. Most techniques require CPU time to form a message before sending it. This is totally unnecessary when using the GOLD RING shared memory networking method for real-time information dispersal.

Before exploring the features of GOLD RING Networking, an interesting trend needs to be addressed. Over the last few years, there have been a number of proposals for “loosely coupled telemetry systems” with strong arguments advocating that there are very few properly designed systems that really need to operate as “real-time” complexes. It is possible that these non-real-time positions have been expressed because of the lag in data movement technology with respect to the needs for it. If the arguments were excuses, they are no longer needed.

GOLD RING is a high performance, state-of-the-art, shared memory network utilizing a (modified) register insertion ring topology. It is designed for fully random memory mapped operations, supporting individual and blocked communication requirements.

The user's real-time tasks involve normal, in-line code operations to "local" memory that is dual ported to the network for replication to all other nodes mapped in as "shared" memory. There are no communication calls or special I/O routines required; simply memory reads and writes. The protocol's overhead is minimized for maximum throughput and minimum. per-node latency to provide the highest performance possible. All protocol overhead, error checking and correction, message driven interrupt and control functions, and access arbitration functions are performed in hardware automatically.

At the system level, all nodes appear to be using the same single memory bank. Each node appears to have immediate (no-contention) access to the shared memory resource because each node has its own private copy of the latest version of the shared memory contents.

Access to the GOLD RING is guaranteed, regardless of bandwidth usage, and is nearly always immediate. There are no tokens or slots to wait for, and there is no master/slave arbitration scheme to contend with. Each node is its own master and is self throttling with all other nodes using the network.

Its use in "real-time" simulation and data acquisition activities can be better presented by example. While many features are left unpresented, the concepts of "real-time" performance in data movement will still be understood.

MAN-IN-THE-LOOP TESTING

"Real-time" experimentation is <u>not</u> the hopeless paradox of times past. While analytical preparations, emulations, and simulations reduce the number and severity of system errors, there will always be a need for experimentation and gathering of "real" empirical information. Conducting "live" tests of subsystem components or complete systems is a very expensive operation and needs to be carefully planned and expertly executed. The more versatile and complex system components become, the more difficult and expensive the testing becomes. Wouldn't it be nice if...

Figure 3-1 presents an experimental unmanned aircraft that is operational as a fully controllable test platform on which new devices can be tested in an environment that may not be safe for a manned vehicle. To be cost effective, efficient use of flight time is mandatory. Therefore, the more experiments that can be performed in one flight, the better. Additionally, if experiments can be modified or aborted based upon results of current testing, much time can be saved for other batteries of tests or shorter flights. If an unforeseen problem arises that can cause catastrophic results if allowed to continue “as is”, “wouldn't it be nice if” the test could be aborted, reconfigured, and rerun while in the same flight?

To perform “live” experimentation safely, inexpensively, and thoroughly, real-time dissemination of tests results is mandatory. Placing the experimenter or artificial intelligence packages in-the-loop for tightly coupled, closed loop experimentation becomes a reality provided the loop performs as a real-time system. This is possible when moving data using the GOLD RING networking architecture.

As an example, assume that the ten computers shown in Figure 3-1 are linked together using 10 meter cables between each of them. A sensor on aircraft #2 indicates that a major catastrophe will occur within 5 milliseconds unless its high-power laser experiment is terminated. It takes 1.2 milliseconds to report that information to a memory location in the “aircraft data acquisition computer”. It takes the “lambda laser experiment computer” 185 microseconds to be interrupted, and then to evaluate the data, perform an abort decision, and issue a command word to a memory location in its computer. It takes another 3.6 milliseconds to upload and execute an abort command once data is written to a memory location in the “digital up-link command computer”. That leaves 15 microseconds for computer-to-computer communications.

Users of “standard” message-passing networks are now grieving for the lost aircraft. Skeptics are saying it can’t be done by any method. GOLD RING can do it on time, every time. From the instant the “aircraft data acquisition computer” writes the sensor state into its shared- memory location until it is available for use at the “lambda laser experiment computer”, 8.591 microseconds will have passed (absolute worst case). Typically this data transit time will be about 4 microseconds, but never more than 8.591 microseconds. It takes another 3.527 microseconds worst case (2 microseconds typical) to move the abort command data from the shared-memory location in the “lambda laser experiment computer” to the “digital up-link command computer” for subsequent experiment termination. Computer-to-computer communications took 6 microseconds (typical), and worst case 12.118 microseconds is still within the operational window for this example. All ten computers were fully aware of the entire chain of events at speeds that equal memory access times.

SPACE STATION COMMUNICATIONS

While this concept is little more than food-for-thought because the time for system design for the Space Station is long gone, it still acts as a catalyst for serious consideration for similar system designs. There are two major classes of information for this type of example, general communications and telemetry information. There are many sources of many forms of data to be provided to many different users.

Most non-GOLD RING local area networking techniques are generalized by the 7 layer model known as the Open Systems Interconnection (OSI) standard, as put forth by the IEEE Computer Society and by the International Standards Organization (ISO). The GOLD RING consists of two quasi-layers; the hardware which automatically performs all of the “physical”, “data link”, “network”, and “transport” functions of the lower 4 layers of the OSI standard model, while reads from and writes to local memory automatically perform all of the “session”, “presentation”, and “application” functions of the upper 3 layers of the OSI standard model. Despite the apparent standard fit, the GOLD RING networking concepts are as different from the OSI standards as night and day.

The trend for the space station’s task of integrating many different forms and sources of data into a coherent universal format is to require all data sources, including telemetry information, to conform to the OSI structure. Some of the penalties for doing this include: the immediate loss of the ability to perform many multi-processor tightly-coupled operations at memory speeds; a dramatic increase in CPU processing requirements and years of software development labor to handle the huge communications tasks; an unnecessarily cumbersome system structure for integrating telemetry data into the standard communications LAN channels; increasingly noticeable lags in information throughput and more frequent “crashes” as more users are brought on-line; the total loss of deterministic performance, and therefore loss of all hope of being a “real-time” communications system.

An alternative technique is to link all computing resources together, regardless of manufacturer or system architecture, and use overlapping shared data pools or a single common data pool information sharing scheme. Every information user will have a local copy of all data, and every local write to shared memory is replicated to all other shared-memory copies at memory (network) speeds. Using this technique, it is possible to perform a real-time update of telemetry sourced data instead of waiting until it is stale and nearly useless while it is packaged into some kind of message format that conforms with the adopted, on-board OSI structure.

Assume that this hypothetical space station has 100 computers all linked together in a single GOLD RING via fiber optics for a total ring distance of 5 kilometers (about 3.1 miles). Any single computer performing a write to shared memory will have successfully changed the contents of that same location in the other 99 computers in less than 104.601 microseconds (absolute worst case); typically, in 55 to 65 microseconds assuming near maximum network loading. This averages out to slightly more than 1 microsecond per computer shared memory write operation (worst case), with typical times at about 1/2 microsecond each; truly networking at memory speeds.

Challenge any other information movement network to come close to those numbers, and behold the results.

EJECTION SEAT TESTING

Figure 4-1 presents a very specific example that is but a sampling of a broad spectrum of possibilities for telemetry-based data acquisition tasks. It also could have been race car or parachute snap or target drone or motorcycle or rocket or pilot support equipment or gun range or vehicle crash or missile guidance or wind blast or experimental aircraft or ... testing. If a class description is needed, it might be those tests requiring the orderly acquisition and processing of large amounts of high-speed information in a hostile test environment, one that forces the use of unconstrained data gathering equipment like PCMs and Telemetry equipment.

The GOLD RING network's concept of large amounts of data at high speeds is access to 8388608 different addressable bytes of information at a throughput rate that equates to a telemetry data stream running in excess of 50 million bits per second; obviously, a greater PCM output rate than normally available. Or, from a PCM/Telemetry source perspective, 128 sensor charmers being converted to 8 bit resolution data being sampled at 10000 samples per second per channel would not use 20% of the available bandwidth on the GOLD RING network.

For this example, assume that the new ejection seat being tested is not only a variable performance intelligent seat with clear-sky-driven guidance and adjustable nozzle thrustors, but it also has radio coupled manual override capabilities, in case difficulties develop during the testing phases of seat development. This positive feedback link requires that data be made available in real-time to the computer system monitoring seat performance for manual override decision and implementation operations.

For the sake of simplicity and high speed, assume that each remote data collection system is linked to the track data center via a one-way, two kilometer (about 1-1/4 miles) fiber-optic cable. Each remote system contains a receiver, a telemetry decommutator, and a computer for storing a backup copy of the data and providing a host platform for the GOLD RING network node that is transmitting the data to the data center. The transit time for each piece of information from when written to shared memory by the decommutation system at the remote data collection system until it is available for use in the shared memory at the track data center is just 11.175 microseconds. Of that total transit time, 10.000 microseconds is required to move the optical version of the information across the fiber optic cable.

At the track data center, the incoming raw data can be sent around a GOLD RING network to online analysis computers for real-time computations such that real- time, in-the-loop override control is possible and the test report is finished (except for the printing) before the dust has settled trackside. The data would be available for system and experiment procedure changes before the system was returned to the hanger following the test. This kind of total system, real-time performance can be reality, right now.

"REAL" and SIMULATED FLIGHTS

Realistic and cost-effective training can be achieved by flying one or a few live aircraft with several simulators, in real-time. A subset of this concept is depicted in Figure 5-1. In-the-air safety is maintained by the total isolation of aircraft operations from the simulation environment, and it is enhanced by subjecting fewer men to dangerous situations per training hour.

The airborne vehicles are outfitted with PCMs and telemetering equipment for providing important flight parameters. The system configuration could include (not depicted) an information link from the range control's radar system for position information. For this example assume that each aircraft's PCM is running at 1000 frames per second, generating 32 (32 bit) words per frame including frame synchronization and error detection/correction information, for an aggregate NRZ throughput of 1024000 bits per second. Not including radio link time, each word of information therefore requires 62.5 microseconds to be serialized on the aircraft and subsequently decommutated for use in parallel format on the ground.

With the simulator running at 100 frames per second, it is obtaining ten times the information necessary to remain linked with the airborne aircraft with which it is flying. While this fact is relatively insignificant, the time from when a particular parameter is measured on an aircraft until the value can be used in a simulation flight equation is crucial. The unachievable ideal of no delay time can be greatly reduced by the GOLD RING network.

The following table analytically provides the deterministic minimum and maximum transit times for two different configurations for this example, differentiated by physical distances. The left half of the table is based upon quasi-long distances between resources (in meters), while the right half is based upon machines more tightly clustered. The maximum node latency of 794 nanoseconds occurs when a message coming into a computer node must wait for network access due to new information being placed (at that same instant) onto the network by the node. The minimum node latency of 247 nanoseconds is achieved when a message coming into a computer node does not have to wait on an outbound message. To achieve the situation where all nodes experience maximum node latency is almost statistically impossible. If all computer nodes are just listening to one data source, then the minimum node latency is experienced and "Tmin" values can be used. Plotting empirical statistical information of minimum through maximum transit times for a variety of network configurations results in a sample distribution curve with a positive Kurtosis and heavy skewing towards the minimum latency time. For the table below, the "Tmax" and "Tmin" values are accumulative from the "real-time data link computer" and are presented in microseconds. Note that the table entries for the termination node at link "I" in the table is the origin, which therefore represents the total network transit time.

LINK:	LENGTH: (in meters)	"Tmax": (in μ seconds)	"Tmin":	LINK:	LENGTH: (in meters)	"Tmax": (in μ seconds)	"Tmin":
A	100	2.289	1.675	A	20	1.889	1.275
B	10	3.133	1.972	B	3	2.698	1.537
C	10	3.977	2.269	C	3	3.507	1.799
D	20	4.871	2.616	D	3	4.316	2.061
E	100	6.165	3.363	E	20	5.210	2.408
F	10	7.009	3.660	F	3	6.019	2.670
G	10	7.853	3.957	G	3	6.828	2.932
H	30	8.797	4.354	H	3	7.637	3.194
I	100	10.091	5.101	I	3	8.446	3.456

As with most systems, many options are available to the system designer for data manipulations and movements. For instance, the "real-time data link computer" can perform sliding window averaging of aircraft data and provide it to the networks shared memory, once per simulator frame time, to minimize network traffic. Additionally, enabling the GOLD RING data filtering will also reduce network traffic to that representing real changes in data. By performing these two operations, a much larger portion of network bandwidth can be used for other communication operations, especially between the four computers tightly coupled to the simulator and with the non-flying human interface workstations' graphic displays.

"Tmin" and "Tmax" times are completely deterministic. "Ttyp" (typical) however is a function of network bandwidth loading and chance, but will always occur in the time window between "Tmin" and "Tmax". For instance, actual data movement time for the radar altimeter reading on aircraft #1 from the "real-time data link computer" to the "instructor station computer" in the clustered model may almost always be less than 3 microseconds, but it will never be more than 6.019 microseconds. Also, note how little time is needed to disseminate real-time information via the network compared to getting the information to the network from the aircraft sourcing it.

CONCLUSIONS

"REAL-TIME" computer communications is: information transit times at memory speeds, with the data arriving error-free and on time every time, with total transparency to the user, regardless of host machine types or numbers or distances between linked machines. Departing from classical message-passing networking techniques (because they can't perform as real-time networks) the GOLD RING replicated shared-memory network redefines all of the necessary attributes for real-time computer communication networks. Its versatility, reliability, and ease-of-use in other market areas has established its place in the real-time computer communication arena. This robust data-sharing technique is a natural candidate for infusion into the many different applications now using telemetry systems, and it can open the door to new areas of use previously not tangible.

REFERENCES

Baker, Donald G. "Local-Area Network with Fiber-Optic Applications" Prentice-Hall, 1986.

Brooks, Tom (Edit.) "The Local Area Network Reference Guide" Prentice-Hall, 1985.

Clark, David D., and Jacobson, Van, and Romkey, John, and Salwen, Howard "An Analysis of TCP Processing Overhead." IEEE Communications Magazine, Vol. 27, Jun 1989.

Fisher, Wayne "Real-time computing demands standards." Computer Design, Vol. 27, Oct 1988.

Freeman, Harvey A., and Thurber, Kenneth J. "Local Network Equipment" IEEE Computer Society Press, 1985.

Glass, Brett "The Token Ring." BYTE Magazine, Vol. 14, Jan 1989.

Gunningberg, Per, and Bjorkman, Mats, and Nordmark, Erik, and Pink, Stephen, and Sjodin, Peter, and Stromquist, Jan-Erik "Application Protocols and Performance Benchmarks." IEEE Communications Magazine, Vol. 27, Jun 1989.

Hawkins, Stephen W. "A Brief History of Time." Bantam Books, 1988,

Heatley, Sharan, and Stokesberry, Dan "Analysis of Transport Measurements Over a Local Area Network." IEEE Communications Magazine, Vol. 27, Jun 1989.

Huber, Daniel E., and Steinlin, Walter, and Wild, Peter J. “SILK: An Implementation of a Buffer Insertion Ring.” The Institute of Electrical and Electronics Engineers, Inc., 1983.

Kauffels, Franz-Joachim, and Johnson, Ian (Trans.) “Practical LANS Analyzed” Ellis Horwood Limited Publishers, (English Edition) 1989.

Lin, Yih-Kang Maurice, and Spears, Dan R., and Yin, Mih “Fiber-Based Local Access Network Architectures.” IEEE Communications Magazine, Oct 1989.

Maxemchuk, Nicholas F. “Twelve Random Access Strategies for Fiber Optic Networks.” IEEE Transactions on Communications, Vol. 30, No. 8, Aug 1988.

McNamara, John E. “Technical Aspects of Data Communication.” Digital Equipment Corporation, 1982.

Nicolo, John M. “Telemetry Bus Linkage with a High-speed Ring Architecture: an approach analysis.” International Telemetering Conference Proceedings, Vol. XXIII, 1987.

Parker, Sybil P. (edit.) “Dictionary of Electronic and Computer Technology.” McGraw-Hill Book Company, 1984.

Pickholtz, Raymond L. (Edit.) “Local Area & Multiple Access Networks” Computer Science Press, Inc. 1986.

Reiss, Leszek “Introduction to Local Area Networks with Microcomputer Experiments” Prentice-Hall, 1987.

Saltzer, Jerome H., and Clark, David D., and Pogran, Kenneth T. “Why a RING?” The Institute of Electrical and Electronics Engineers, Inc., Proceedings of the Seventh Data Communications Symposium, 1981.

Shelly, Larry R. “The Real-time Telemetry Processing System III.” International Telemetering Conference Proceedings, Vol. XXIII, 1987.

Singh, Inder, Dr. “Real-time Needs Standards.” Motion Magazine, Vol 7, No. 2, Mar/Apr 1991.

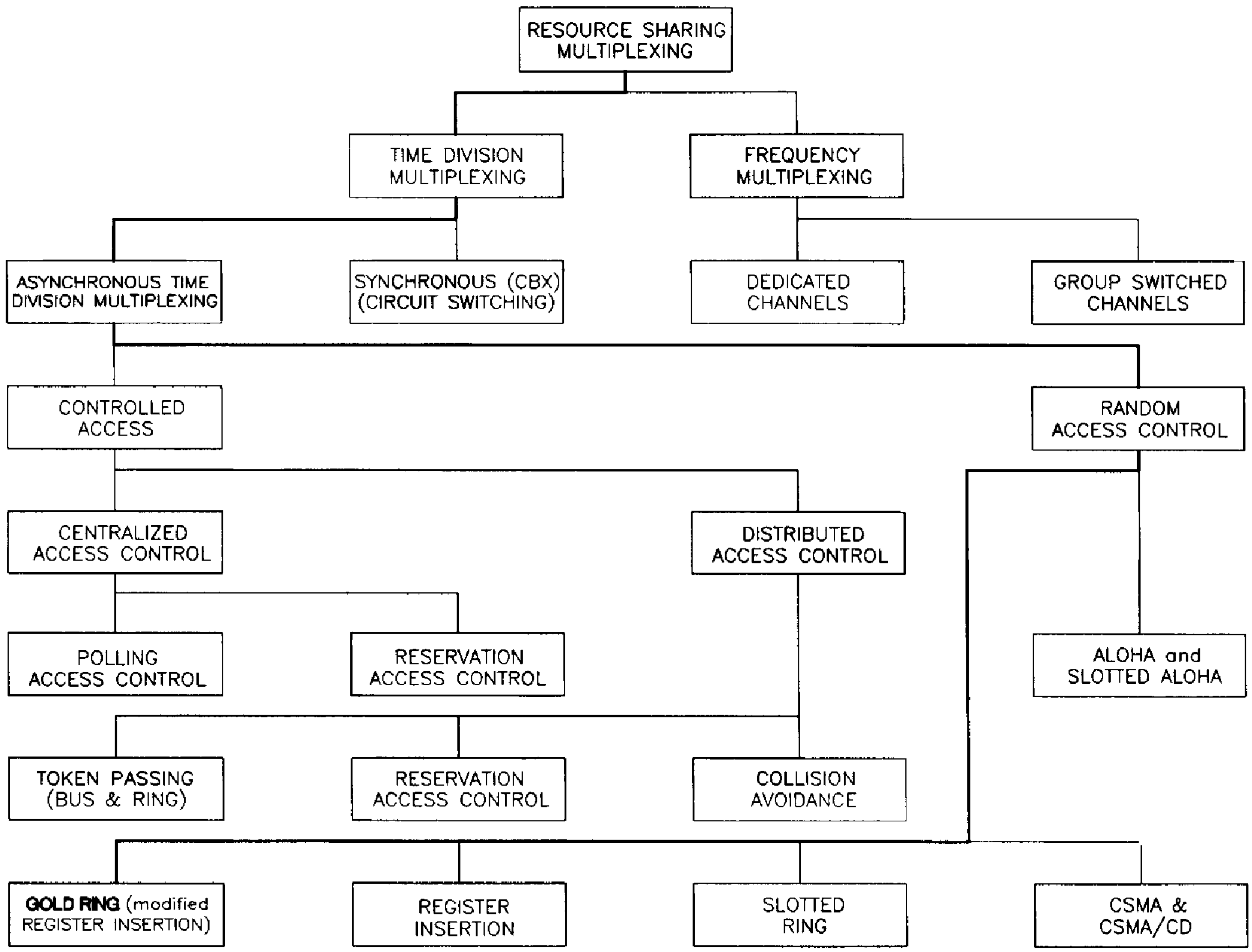
Skov, Morten “Implementation of Physical and Media Access Protocols for High-Speed Networks.” IEEE Communications Magazine, Vol. 27, Jun 1989.

Stallings, William, Ph.D. “Data and Computer Communications.” Macmillan Publishing Company, 1985.

Stallings, William, Ph.D. “Tutorial Local Network Technology” IEEE Computer Society Press, 1983.

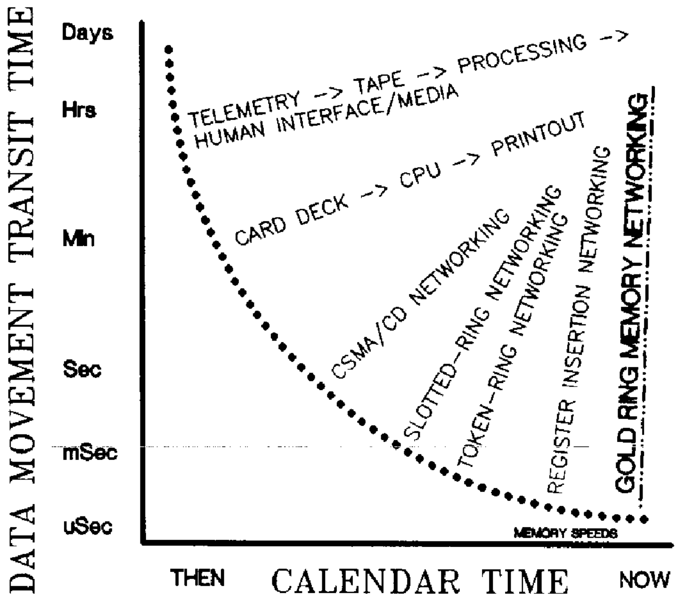
Straehley, Erwin H. “The Case for Non Real-time System Integration.” International Telemetering Conference Proceedings, Vol. XXIII, 1987.

White, V. E. “Network Interfaces and Protocols.” Network Management Services, 1985.



RESOURCE SHARING MULTIPLEXING
TIME DIVISION MULTIPLEXING
FREQUENCY MULTIPLEXING
ASYNCHRONOUS TIME DIVISION MULTIPLEXING
SYNCHRONOUS (CBX) (CIRCUIT SWITCHING)
DEDICATED CHANNELS
GROUP SWITCHED CHANNELS
CONTROLLED ACCESS
RANDOM ACCESS CONTROL
CENTRALIZED ACCESS CONTROL
DISTRIBUTED ACCESS CONTROL
POLLING ACCESS CONTROL
RESERVATION ACCESS CONTROL
ALOHA and SLOTTED ALOHA
TOKEN PASSING (BUS & RING)
RESERVATION ACCESS CONTROL
COLLISION AVOIDANCE
GOLD RING (modified REGISTER INSERTION)
REGISTER INSERTION
SLOTTED RING
CSMA & CSMA/CD

Figure 1–1



DATA MOVEMENT TRANSIT TIME
Days
Hrs
Min
Sec
mSec
uSec
TELEMETRY -> TAPE -> PROCESSING ->
HUMAN INTERFACE/MEDIA
CARD DECK -> CPU -> PRINTOUT
CSMA/CD NETWORKING
SLOTTED-RING NETWORKING
TOKEN-RING NETWORKING
REGISTER INSERTION NETWORKING
GOLD RING MEMORY NETWORKING
MEMORY SPEEDS
THEN
CALENDAR TIME
NOW

Figure 2-1

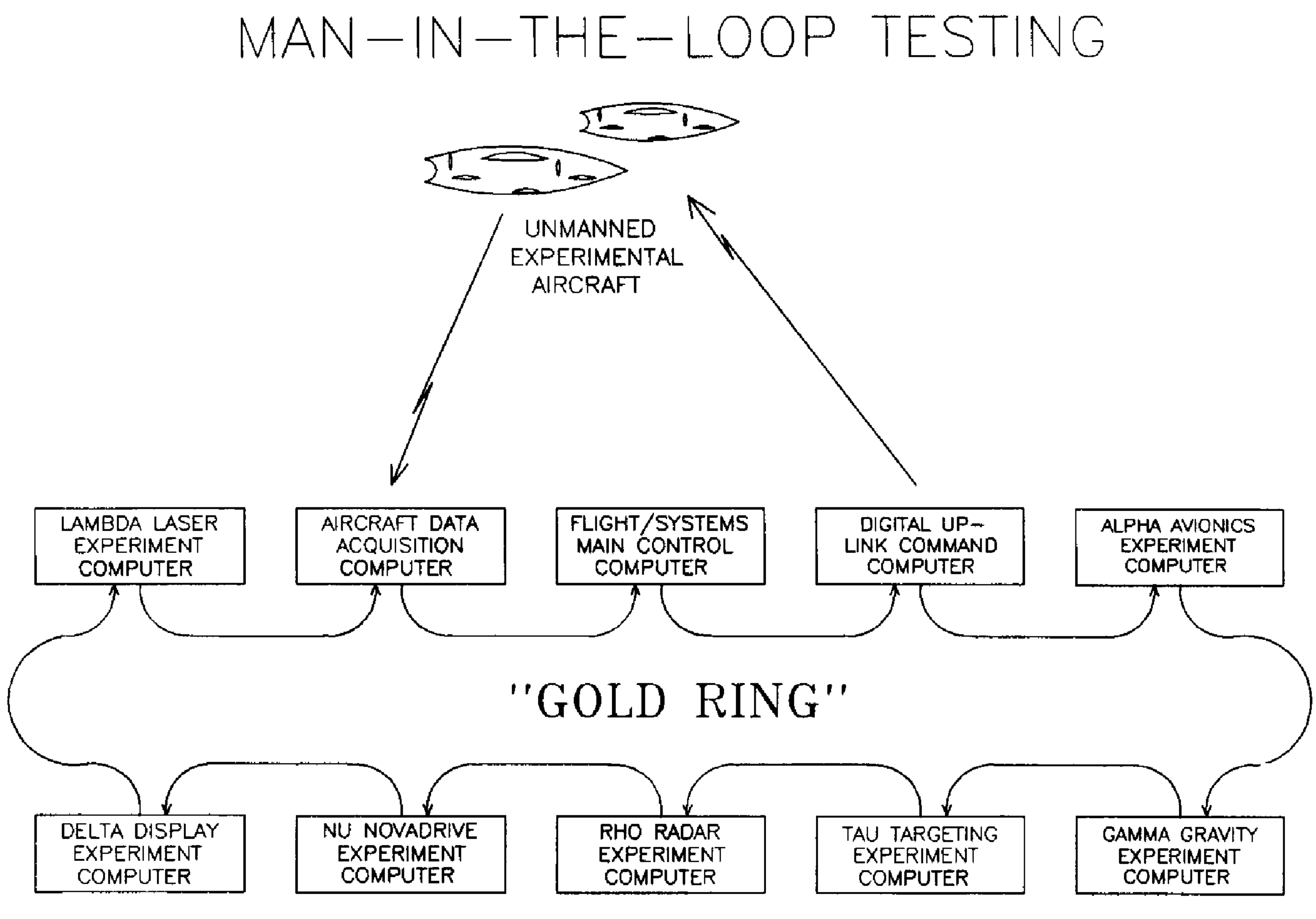

MAN-IN-THE-LOOP TESTING
UNMANNED EXPERIMENTAL AIRCRAFT
LAMBDA LASER EXPERIMENT COMPUTER
AIRCRAFT DATA ACQUISITION COMPUTER
FLIGHT/SYSTEMS MAIN CONTROL COMPUTER
DIGITAL UP-LINK COMMAND COMPUTER
ALPHA AVIONICS EXPERIMENT COMPUTER
"GOLD RING"
DELTA DISPLAY EXPERIMENT COMPUTER
NU NOVADRIVE EXPERIMENT COMPUTER
RHO RADAR EXPERIMENT COMPUTER
TAU TARGETING EXPERIMENT COMPUTER
GAMMA GRAVITY EXPERIMENT COMPUTER

Figure 3-1

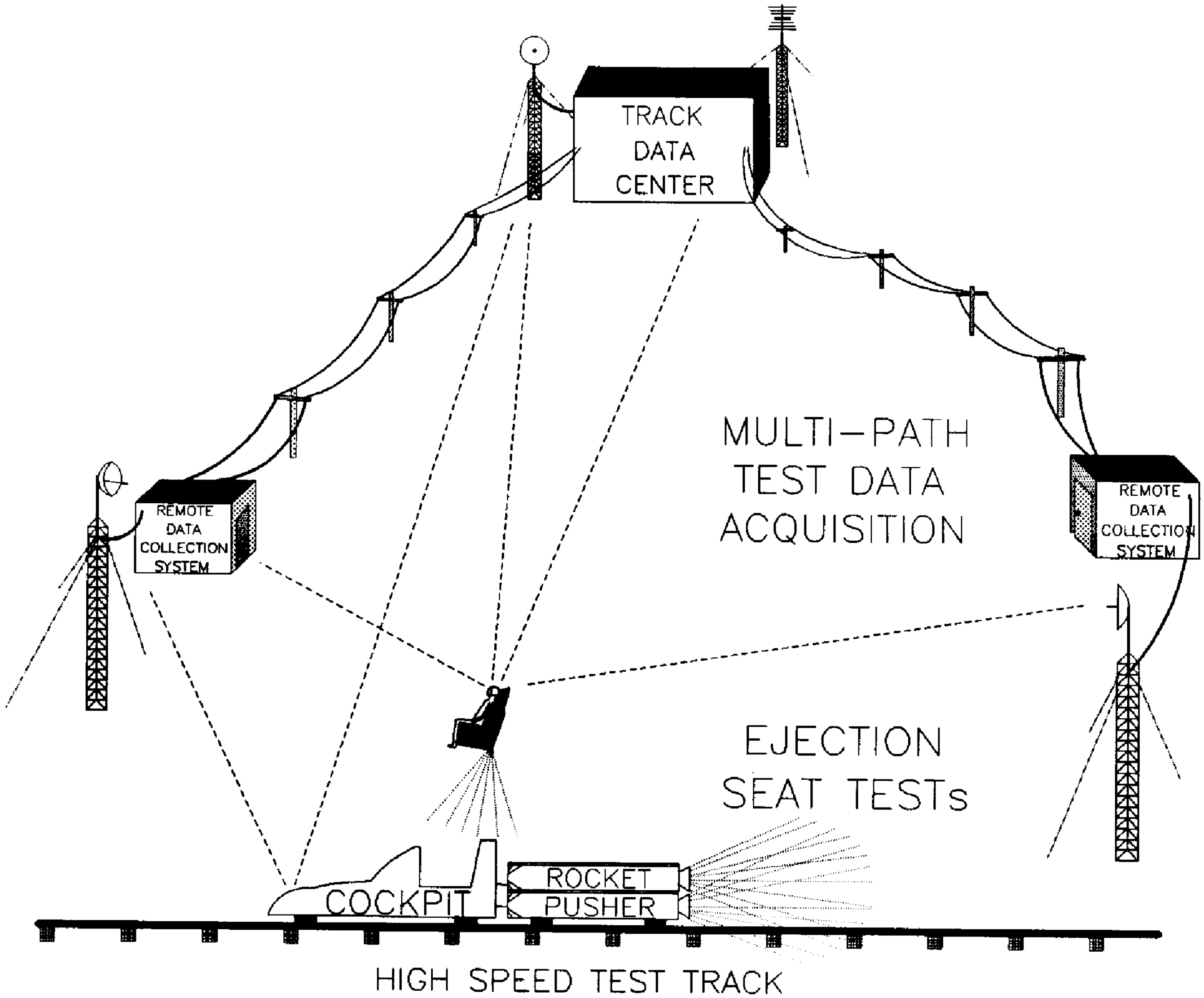

TRACK
DATA
CENTER
REMOTE
DATA
COLLECTION
SYSTEM
REMOTE
DATA
COLLECTION
SYSTEM
MULTI-PATH
TEST DATA
ACQUISITION
EJECTION
SEAT TESTs
COCKPIT
ROCKET
PUSHER
HIGH SPEED TEST TRACK

Figure 4-1

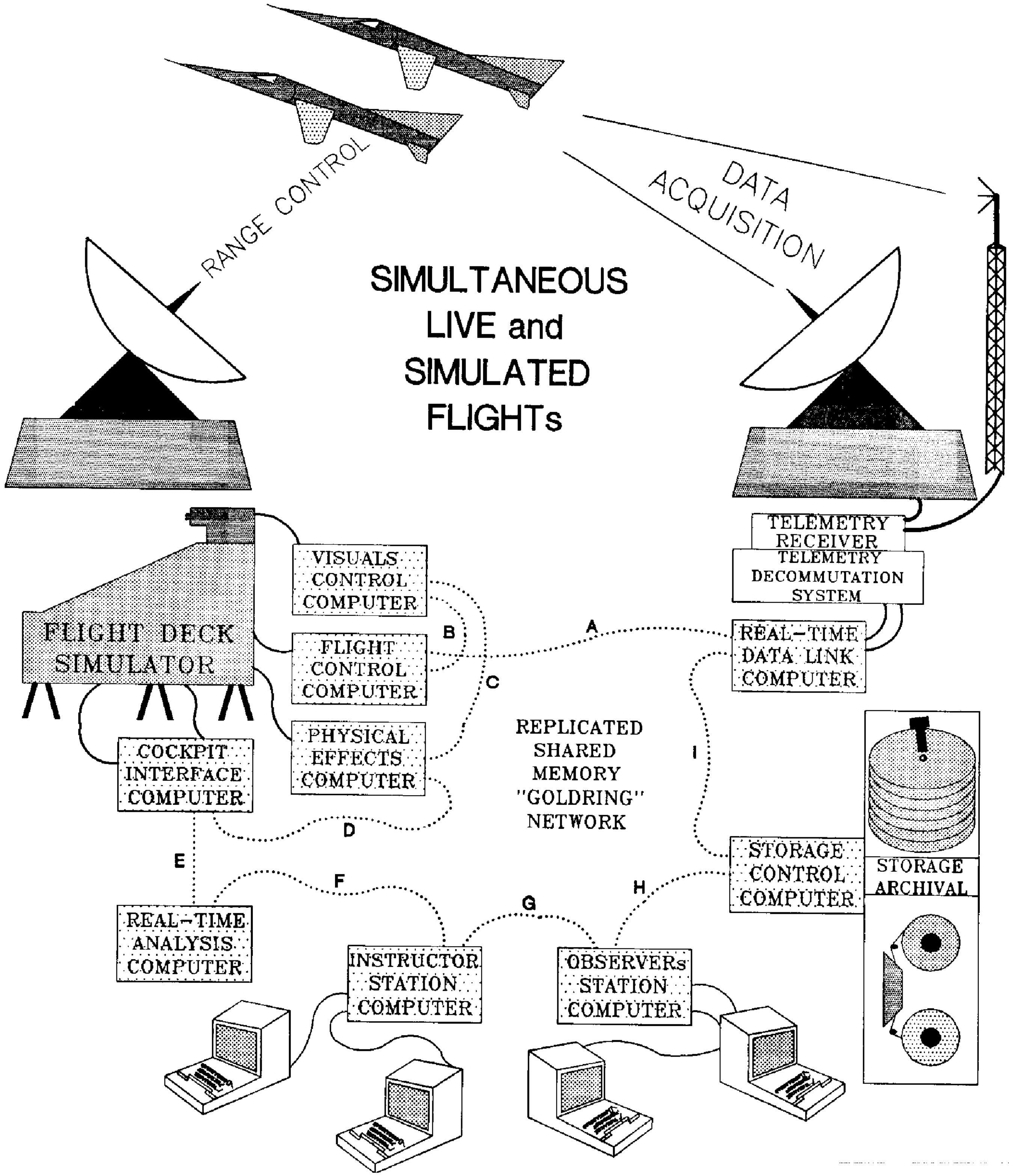

RANGE CONTROL
DATA ACQUISITION
SIMULTANEOUS
LIVE and
SIMULATED
FLIGHTs
TELEMETRY RECEIVER
TELEMETRY DECOMMUTATION SYSTEM
VISUALS CONTROL COMPUTER
FLIGHT DECK SIMULATOR
FLIGHT CONTROL COMPUTER
REAL-TIME DATA LINK COMPUTER
A
B
C
PHYSICAL EFFECTS COMPUTER
COCKPIT INTERFACE COMPUTER
REPLICATED
SHARED
MEMORY
"GOLDRING"
NETWORK
I
D
E
F
G
H
STORAGE CONTROL COMPUTER
STORAGE ARCHIVAL
REAL-TIME ANALYSIS COMPUTER
INSTRUCTOR STATION COMPUTER
OBSERVERs STATION COMPUTER

Figure 5–1