DISTRIBUTED SYSTEMS DESIGN USING SEPARABLE COMMUNICATIONS

by

A.C. Capel and G. Yan

ABSTRACT

One of the promises of distributed systems is the ability to design each process function largely independently of the others, and in many cases locate the resulting hardware in close proximity to the application. The communications architecture for such systems should be approached in the same way, using separable communications facilities to meet individual sets of requirements while at the same time reducing the interactions between functions. Where complete physical separation is not feasible and hardware resource sharing is required, the protocols should be designed emphasizing the logical separation of communication paths. This paper discusses the different types of communications for process control applications and the parameters which need to be characterized in designing separable communications for distributed systems.

DISTRIBUTED SYSTEMS DESIGN USING SEPARABLE COMMUNICATIONS

by

A. C. Capel and G. Yan

1. DISTRIBUTED ARCHITECTURES FOR PROCESS CONTROL

To achieve an acceptable balance between cost and performance in designing distributed systems, architectures must be selected to match the specific requirements of each application. Although no practical design methodology is yet available to help designers to better understand distributed system properties and to guide them in generating more accurate specifications, a generalized design sequence was formulated to illustrate the application of established design methods in three major design areas: processing clusters, data communications, and databases [1].

In the data communications area, a wealth of information is already available [2,3]. Much of this work is focussed on the business data processing market where resource sharing and remote human interactive services [4,5] must be supported over large physical distances, using facilities based on existing plant. Other work, directed towards the sharing of in-house computing, storage and terminal equipment, has led to highly-multiplexed localized single bus networks [6,7,8]. In contrast, the process control requirements, such as those found in a nuclear power plant, generally encompass a spectrum of communication needs which will call for a range of solutions.

One of the promises of distributed systems is the ability to design each process function largely independently of others, and in many cases locate the resulting hardware in close proximity to the application. Consequently, effective communications must be provided since functions now communicate over larger physical distances and between many generically dissimilar machines. Rather than following the trend

of using single highly multiplexed buses, it is proposed that separable communications would best meet the requirements of process control applications.

2. SEPARABILITY AS A DESIGN OBJECTIVE

A distributed system can be envisaged as a complex assembly of hardware/software elements overlayed by an interconnected assembly of data acquisition, processing, and control functions. Each of these functions, while performing the individual actions of, for example, pressure control, temperature control, operator input, should be designed in a manner which reduces interactions between each other due to implementation details.

To simplify the design tasks, the general approach taken is to subdivide the total information transport system into subsystems, each carefully matched to a particular set of requirements. A full understanding of the requirements is essential so that the partitioning of the total job can be carried out efficiently.

This partitioning is similar to that done for the components of the distributed system itself. Figure 1 illustrates this analogy of equating centralized processing with highly multiplexed communications, and distributed processing with separable communications. The rationale is simple. Instead of simply looking at the cost savings of shared hardware, which is the major attraction for both centralized processing and highly multiplexed communications, one should now consider the better matching of requirements with the available electrotechnologies which would lead to distributed architectures and separable communications.

While some functions with stringent requirements will use separate communications hardware, clearly the cost and mechanical constraints of today will require some sharing of common equipment. The first task is to identify user groups which have conflicting requirements or those for which the sharing of common facilities is not desirable nor economically attractive. Within each sharing user group, designs which make one user logically independent of the others should be adopted. In this manner certain performance parameters, e.g. throughput, can be guaranteed to be independent of the state of other users.

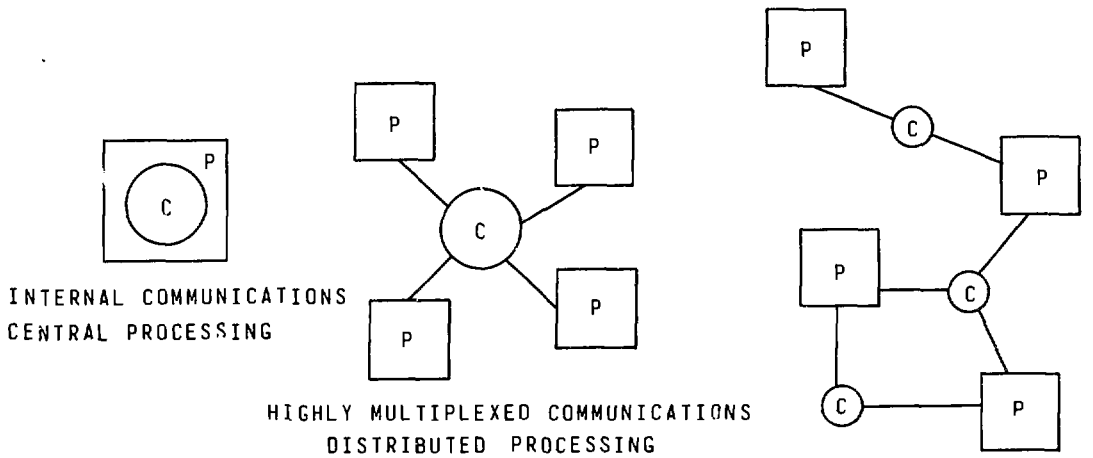

FIGURE 1: COMPARISON OF PROCESSING AND COMMUNICATION ARCHITECTURES

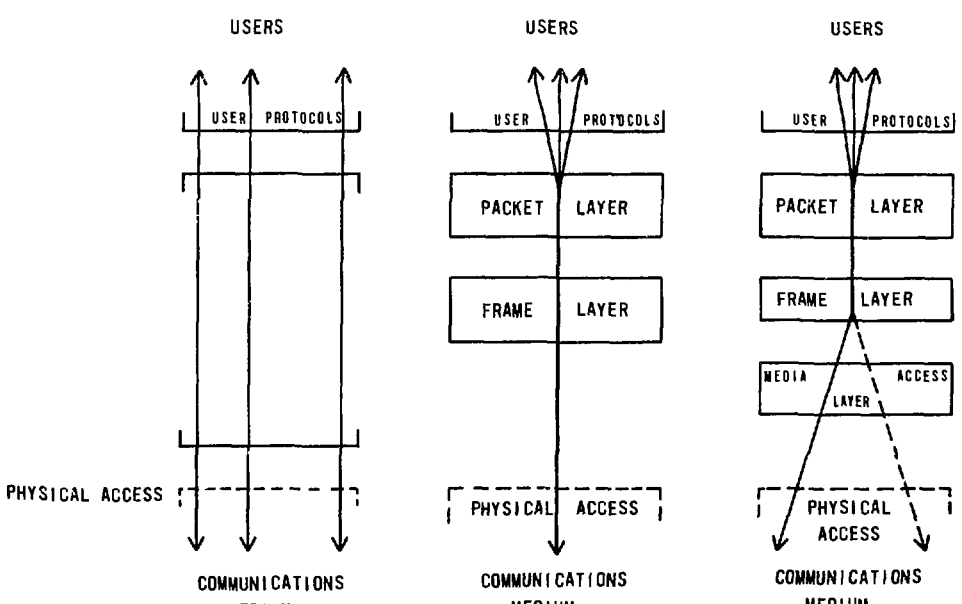

FIGURE 2: SEPARABLE PROTOCOLS

2.1 Physical Separability

Physical separability pertains to the use of totally distinct hardware for different communication functions. Unique requirements call for separate subsystems and the extent of this is limited by the cost and physical complexity of the resulting design. In REDNET [9], a unique requirement for a broadcast time base for all processing elements led to the provision of a time-of-day subsystem which uses separate hardware. On the other hand, a number of multipoint-to-multipoint subsystems each providing identical but separate inter-function communications are overly costly and complex, and the best approach may be a single multiplexed subsystem. This option was selected for terminals and some inter-process communications in REDNET [10].

Clearly a balance must be struck between the current cost savings of shared hardware, with attendant increases in interaction between functions, and the advantages of physically separate facilities.

2.2 Protocol Separability

Even when communications hardware is to be shared, the design of protocols can be carried out in a manner which emphasizes the separability of support to individual users. Since it is the protocols which permit resource sharing in the first place, it is in the design of protocols that separability must be entrenched. This aspect of protocol design is one not generally considered in the available literature.

Currently the "layered" approach to protocol design is favoured [11] and three examples are shown diagrammatically in Figure 2. Each one is designed to interface a number of users to information transport equipment. Every layer of each protocol operates as independently as possible of the other layers, while progressively insulating the user from the specific considerations of the hardware transmission media. Clearly, for shared media, separability of individual user traffic becomes progressively less obvious at the "lower" layers.

Figure 2(a) shows how physically separate media promote the use of completely separable protocols, although care must be taken when using common processing hardware and software. Separate hardware does one no good for example, if a poorly controlled shared buffer pool is used.

The CCITT X.25 [12] protocol of Figure 2(b) combines logical channels at the Packet Layer so that they may be subjected to common controls at the Frame Layer. For X.25, data flow and error control procedures are available at both layers, although their use at the Frame Layer without adequate protection at the Packet Layer, will allow logical channel data flows to potentially interfere with one another.

Figure 2(c) represents the internal INTRAN structure used for the REDNET terminal support subsystem [10]. Since this subsystem uses a multi-access media, an extra Protocol Layer is added to control media access. Additionally some units have access to a second physical channel (for high data rate transmissions), and so a split in the data flow is made at the lower layers to route the data appropriately. (The second channel utilizes a different media access protocol more suited to large data transmissions.)

The INTRAN design attempts to minimize user interdependence. All flow control procedures reside at the Packet Layer, with none provided at the Frame Layer where potential common blocking might occur. Similarly since ARQ* error control procedures are used, and since these procedures may also cause blocking, error detection only is provided at the Frame Layer with recovery provided at the Packet Layer only.

Similar measures are taken at the Media Access Layer but in addition, this layer must take into account the operation of other units attached to the multi-access media. INTRAN uses a primary channel access technique which guarantees a minimum level of service irrespective of other loads. Additional capacity is available depending upon total load. The optional second channel uses a different access mechanism which is more efficient in channel usage but provides no access guarantees beyond strictly statistical ones.

---

*ARQ: Automatic request for retransmission requires retransmission of erroneously sent data; as opposed to FEC - forward error control - procedures which include error correction codes within the initial transmission.

For communications systems which use store and forward or inter-linking intermediaries, layers of protocol will exist between users other than those shown in Figure 2. Considerations similar to those already discussed should be applied in these cases.

3. COMMUNICATION TYPES

Before a separable communications facility can be designed, it is necessary to identify the different communication types to be supported, which can be grouped into: machine-to-machine, man-to-machine, and man-to-man communications. Although important in practice, man-to-man communications will not be dealt with further.

3.1 Machine-to-Machine Communications

Five examples are analyzed and summarized in Figure 3 to delineate a set of parameters needed to characterize the communication types. For specific designs, the identification of communication types is derived directly from the requirements.

3.1.1 Inter-Process Communications (IPC)

Refers to the interchange of short, concise coordination messages between functions (or tasks). Large data transfers are specifically excluded and relatively low throughout requirements are expected. These interchanges are subject to very critical constraints in terms of the control of transmission delays since task semaphoring schemes (using IPC) are very sensitive to execution delays. Uniform transaction arrival rates can be expected leading to an even loading on the communications facility.

For distributed systems where some functions may execute in more than one physical machine, addressing by function would be an asset.

3.1.2 File Transport

Refers to all interactions between processors and/or storage devices involving the movement of large amounts of data. These data can be characterized in terms of the length of time for which they remain valid. Thus, rapidly changing data must often be transported with correspondingly short delay times while slowly changing data can tolerate longer delay times.

		DATA FORMAT UNIT OF DATA	TRANSMISSION DELAY	THROUGHPUT	LOAD RANGE	DATA FLOW CONTROL (buffering)	ERROR CONTROL	ADDRESSING	TOPOLOGY	PHYSICAL ENVIRONMENT
MACHINE-MACHINE	IPC	medium	short predictable	low	even (bursty)	no	FEC	by function	MM	uniform "good"
	TRANSPORT FILE BATCH -LIKE	large	non-critical	high	well distributed	yes	ARQ	by function	MM	uniform "good"
	TRANSPORT FILE REAL -TIME	large	predictable	high	bursty	yes/no	ARQ/FEC	by function	MM	uniform "good"
	SENSOR	small	short predictable	low	even	no	FEC/none	physical	PP PM	harsh
	ACTUATOR	medium	short predictable	low	bursty	no/yes	FEC	physical	PP MP	harsh
MAN-MACHINE	TERMINALS STATUS REPORT	medium -large	non-critical	medium	bursty	yes/no	none	by function	MM PM	variable "good"
	TERMINALS COMMAND CONTROL	medium	short	low	bursty	no	higher level	by function	MM	variable "good"
	DIRECT	small	short	low	bursty	no	FEC	physical	PP PM	harsh

Error Control:
- FEC - Forward Error Control (error correcting codes)
- ARQ - Automatic Repeat Request (error detect + retransmission)

Topology:
- MM - Multipoint-to-Multipoint
- MP - Multipoint-to-Point
- PM - Point-to-Multipoint (broadcast)
- PP - Point-to-Point (direct wire)

FIGURE 3: SUMMARY OF COMMUNICATION TYPES AND REQUIREMENTS

In Figure 3 a more detailed characterization has been made with the extremes represented by "batch-like" and "real-time" descriptors. Real-time data are moved to real-time functions which rely on short transmission delays. Batch-like data can be subjected to transmission delays since the functions using these data are subjected to execution delays due to queuing and other factors.

Batch-like data will also require extra data flow control procedures since batch functions are less likely to be ready to accept the data at the instant of arrival. Loading of the communication subsystem will be more even since the higher tolerance on transmission delays will allow longer averaging periods and thus lower peak loads.

3. .3 Sensor and Actuator Communications

Involve the transfer of information between real-world interfaces and control functions. Traditionally, these devices use special purpose interface and communication subsystems. In a distributed system, where such data may be moved to many physical locations, such communications must be considered to be an integral part of the total information system.

It is difficult to identify communications requirements for sensors and actuators without a knowledge of their functional partitioning. For example, a undirectional broadcast-style subsystem could be provided if sensor calibration functions are partitioned away from the sensor. If sensors are capable of responding to specific commands, bi-directional communication is required.

Generally, devices to be served are geographically distributed, oftentimes have highly periodic data throughput requirements, and require predictable (and possibly short) transmission delays. These requirements can be judged in more detail based on the control algorithms in use.

In many cases the physical environment of sensors and actuators vary widely. Thus the communications subsystem must be suitable, in terms of cost, wiring technique, etc. for the range of such environments.

3.2 Man-to-Machine Communications

Communications with man encompasses a wholly different range of constraints, both in the area of timing and presentation of data, and in the area of variability of input. A demand for a display, for example, may occur at any time and the required system response speed is based on difficult-to-determine psychological factors. Man communications also precludes the use of complicated protocols to control data flow and errors.

3.2.1 Status Reporting

Status reporting is likely to be implemented as a "read-only" function. Specific reports may be elicited by operator demand or general information may be presented by "broadcast" displays. Displays may originate from one database or may be made up from information obtained from several different sites. These sites may be direct outputs from sensors, functions, controllers, and databases. Combining data into a single display could be done dynamically according to total system state or operator enquiry and may be a local function of an "intelligent" display unit.

3.2.2 Command Control Communications

Control operations are typically "read-write" functions accessible by selected personnel which allow them to change plant operating parameters. Status and command control functions may be combined in one device, although additional security would be required and personnel access procedures become complicated.

The command control sequence is: select the function, pass any security control information (key) to the function and make a secure transmission of the new parameter settings. While a variety of conventional techniques can be used to secure the operator interface to the control facility, security within the communications facility is essential.

3.2.3 "Direct" Man Communications

"Direct" communications between man and the equipment pertains to functions which require minimal intelligence at the man/machine interface. Such automatic functions

as: door lock switch sensing and (possible) activation, personnel presence sensors, alarm annunciation are included. Typical data flows for such applications have high priority, are bursty, but have low average data rate requirements.

4. CHARACTERIZATION OF COMMUNICATION REQUIREMENTS

4.1 Data Format and Units of Data

Many communication parameters are defined assuming a specific data format and are based on the minimum quantity of this data that must be transferred at any one time. Terms such as bits, bytes, words, records, blocks, and files may be used to describe the users' data format. The concept of a "unit of data" is helpful to describe that quantity of data which, when passed through the communications system, is sufficient to initiate significant processing at a receiver. A unit of data might range from a binary indicator of a contact closure state to a whole file which describes a matrix of temperature readings. One can envisage that a communications scheme for the scanning of sensors (individual small units of data) would be different than that for large file transfers.

4.2 Transmission Delay

An event or condition at one location is reported to another via a communications system which introduces a transmission delay. Delays between sensor inputs and calculated control outputs are very important parameters in control system design. Some control algorithms can be constructed to compensate for known delays, but in general one would like to have delays which are as small and as predictable as possible. Interprocess communications is very susceptible to transmission delays. A function in one place requesting a service of another is often subjected to time coordination problems. In some cases transmission delays can have an excessive impact on task semaphores.

The delay will be the sum of several factors including propagation delay, service and waiting times, and delays introduced by error control procedures. The latter two can be significant for process control applications. When sharing common hardware, access to the equipment must be made prior to transmission and this will affect the waiting time. ARQ error control procedures may require several attempts before transmission is successfully completed.

4.3 Throughput and Load Range

The communications throughput requirements are determined by the amount and speed of the data to be transported. In a shared facility, the delay imposed on a transfer will vary due to loading and the throughput available to each user must be specified in terms of the varying load to be expected. During certain system states, variations can be large: trip, shut-down, start-up, etc. Occasional large instantaneous (worst case) loads can also be imposed even during normal operation if periodic data transport is unsynchronized.

When discussing loading, users must estimate the loads which they expect to impose (and when) on the communications equipment. The communications designer must clearly indicate guaranteed and average performance in the light of all user loads. In many cases, guaranteed performance levels will be an important factor in determining whether sharing of the communications subsystem is feasible.

4.4 Error Control

While every communication channel is susceptible to errors, the specification of error rates must be made in context since unrealistic demands often lead to inefficiencies which do not meet real overall performance goals. Sensors and human operators, for example, often have a high error (failure) rate for which processing functions make due allowance. Communications between process and actuator on the other hand are more stringent and require considerably more attention. Various error control techniques are well-known and they must be selected with a knowledge of the requirements of the data being transported.

For example, forward error correcting codes allow the receiver to correct all known errors without further interaction with the data source. Unfortunately, error correcting codes require greater data transmission overheads than simpler error detection-only codes. Automatic repeat request (ARQ) procedures, which use error detection-only codes, require that the sender maintain copies of all unacknowledged messages until they have been received correctly.

4.5 Flow Control

Flow control procedures are used to co-ordinate the transmission and reception of data. Flow control is not always required, e.g. periodically produced sensor scanning for which the receiving process needs only the "latest" value. Other situations, (e.g. terminal displays for operations staff), cannot guarantee that data source, data transmission media, and data sink can be made available at the same time. Flow controls may be required to limit loading during critical times, and may also be integrated with error control procedures.

4.6 Topology

Considering the flow of data, four basic topologies can be described. A multipoint-to-multipoint configuration allows a number of data sources to communicate with data sinks with interconnection patterns changing dynamically. A point-to-multipoint link has a single data source which broadcasts to a number of sinks. Multipoint-to-point configurations have the opposite topology, e.g. sensors scanned from a single point. The point-to-point configuration permits two devices to communicate with one another only.

It is clear that vastly different error control procedures, protocols, etc. are applied to the different topologies. A broadcast topology can be made very secure for example, simply by ensuring that receivers cannot transmit, although FEC rather than ARQ error control would be required.

4.7 Addressing

If functions are permitted to change their physical addresses dynamically, then procedures must be provided to allow the communications system to locate them. A relocated function must tell the system its current address and these interactions if dynamic, can become very complex.

4.8 Physical Environment

It is evident that the basic components of communication subsystems serving each type of user must be appropriate for the physical environment. For example,

sensors and actuators must be connected to communications equipment appropriate to local conditions. Requirements placed on computer-to-computer equipment are likely to be less stringent.

5. CONCLUSIONS

A uniform approach using separability as a design objective is effective in developing communications facilities for real-time process control applications. Parameters, characterizing the different types of communications, have been discussed and these can be used as criteria in selecting the communications subsystems for specific applications. The use of separable communications is consistent with the functional partitioning strategy for system architectures and is another step towards realizing the advantages of distributed systems.

6. REFERENCES

1. L'ARCHEVÊQUE, J.V.R., YAN, G., *"On the Selection of Architectures for Distributed Computer Systems in Real-time Applications"*, IEEE Transactions on Nuclear Science, NS-24, pg 454-459, February 1977.

2. GREEN, P.E., LUCKY, R.W., *"Computer Communications"*, IEEE Reprint Series, 1974.

3. FALK, G., McQUILLAN, J.M., *"Alternatives for Data Network Architectures"*, IEEE Computer pg 22-29, November 1977.

4. GREEN, W., POOCH, V.W., *"A Review of Classification Schemes for Computer Communication Networks"*, IEEE Computer pg 12-21, November 1977.

5. SCHWARTZ, M., BOORSTYN, R.R., PICKHOLTZ, R.L., *"Terminal-Oriented Computer-Communication Networks"*, Proceedings IEEE, Vol. 16, n 11, pg 1408-1423, November 1972.

6. FARBER, D.J., et al, *"The Distributed Computing System"*, IEEE Compcon 73 Digest, pg 31-34, 1973.

7. METCALFE, R.M., BOGGS, D.R., *"Ethernet Distributed Packet Switching for Local Computer Networks"*, Communications ACM, Vol. 19 n 7, pg 395-404, July 1976.

8. WATSON, R.W., *"Network Architecture Design for Back-end Storage Networks"*, IEEE Computer pg 32-48, February 1980.

9. YAN, G., L'ARCHEVÊQUE, J.V.R., WATKINS, L.M., *"Distributed Computer Control Systems in Future Nuclear Power Plants"*, Nuclear Power Plant Control and Instrumentation Vol. II, International Atomic Energy Agency, Vienna, 1978.

10. SHAH, R.R., CAPEL, A.C., PENSOM, C.F.,*"Distributed Terminal Support in a Data Acquisition System for Nuclear Research Reactors"*, IWG/NPPCI Specialists' Meeting on Distributed Systems for Nuclear Power Plants, International Atomic Energy Agency, May 12, 1980.

11. WALDEN, D.C., *"The Evolution of Host-to-Host Protocol Technology"*, IEEE Computer pg 29-38, September 1979.

12. *"Provisional Recommendations X.3, X.25, X.28 and X.29 on Packet-Switched Data Transmission Services"*, International Telegraph and Telephone Consultative Committee (CCITT), Geneva, 1978.