Software Engineering of the Micro/Mini Computer Subnet in Computer Networks

Schneidewind, Norman F.

Computer Software and Applications Conference, 1978, COMPSCAC '78, The IEEE Computer Society's Second International Conference. pp. 508-513
http://hdl.handle.net/10945/45284

Downloaded from NPS Archive: Calhoun
A software design of a micro/mini subnet of a campus computer network is described. Software engineering aspects of this design include: identification of terminal, data management and communications functions which are appropriate for micro/mini implementation; logical and physical placement of micro/mini facilities; central versus autonomous operating system control; number and type of protocol layers. The practicality of distributing the above functions at the micro/mini level in a computer network is assessed.

Introduction

With advances in small computer technology and the concomitant diseconomies of scale which can occur in centralized, large-scale computer operations, organizations are increasingly looking to a more decentralized form of computing to satisfy their processing needs. One form of decentralization is distributed processing. Our purpose is to describe a methodology for the functional and software design of the micro/mini subnet of a distributed computer network. First we describe computer network characteristics and structure, contrasting distributed networks with centralized networks. This is followed by a description of the design methodology. Next the methodology is applied to the preliminary design of a campus micro/mini subnet. Finally, some conclusions are offered concerning the practicality of distributing certain functions at the micro/mini level in a computer network.

Unfortunately, there is a lack of agreement concerning the definition and characteristics of distributed processing. Differences of opinion on technical and semantical points obfuscate discussions regarding the merits and applicability of distributed systems. Some of the disagreement stems from an insistence on requiring all network functions to be distributed in order for a network to qualify as a distributed system. The types of networks to be described would not qualify as distributed by that definition. However, since the systems possess a number of distributed properties and are fundamentally different from centralized networks, the distributed network classification seems appropriate. These properties are the following:

- More than one processing node.
- More than one communications node.
- Several levels of processing complexity.
- Interconnection of nodes at the same and different levels.
- In certain cases, assignment of a processing or data base function to more than one processing node.
- More than one autonomous operating system, each of which executes in a different node.

Computer Network Characteristics and Structure

Computer networks possess both logical and physical characteristics. The former is exemplified by the functions and services provided by the network, especially in terms of the user view of the network, while the latter is illustrated by number and types of nodes, network topology, geographic location of nodes and channel bandwidth. A logical characteristic which exerts considerable influence on network performance and reliability is the degree of autonomous control exercised in the network over communications, data base management, language translation, program execution and resource assignment. The opposite ends of this spectrum are centralized control at a single node versus control which is distributed across several nodes. This characteristic, along with the logical entities, i.e. processes and process protocols, define the logical structure of the network. The physical structure of the network, consisting of node and link capacities, and network topology are derived from the logical structure. Although logical and physical structures may be analogous, this is not always the case. For example, a network could operate with a centralized control located at a single node, but the physical structure of the network could be highly distributed on both a functional and geographical basis. The converse can also be true in that a single computer installation at one location could operate with distributed processing and data base functions across several hardware components within the one complex. It should be noted that certain logical functions and resources may be centralized, such as control, while others are distributed, for example, processing and data management.
Another logical characteristic which has a physical analog is the degree of heterogeneity of the network. This is the extent to which various nodes and links have different characteristics as a result of performing different functions. Physically, nodes may differ as to hardware specifications, operating system characteristics or vendor. It is not always the case that variety of logical function requires heterogeneity of physical properties. The hardware in such a network may be identical for reliability and maintainability purposes. On the other hand, nodal logical function may vary and the hardware employed at the nodes may vary in accordance with logical function requirements. In the design of computer networks, a mapping must be made between desired logical functions and appropriate physical components. This process is shown in Table 1 for centralized and distributed networks.

Much of the computer network literature is based on the analysis of networks which have already been specified, either on the basis of existing (e.g., ARPA) or hypothetical networks. While these articles have increased our understanding of network characteristics and operations, the more fundamental problem, and a subject of this paper, is the specification of the network in the first place. A methodology for designing computer networks will be described.

**Design Methodology**

Logical design of computer networks involves, in large measure, the identification of functions and services to be performed by the network and the subsequent assignment of functions to nodes and links. Physical design involves the assignment of capacities to nodes and links. Capacities consist primarily of processing speed, storage size, access time and data rate. The general flow of the design methodology logic is shown in Figure 1. The design methodology is applied to system objectives to produce network alternatives which are subjected to performance, cost and reliability evaluations. The approach which will be used in this paper is to sketch out the general procedure, without reference to a specific application. This will be followed by applying the methodology to the design of the micro-mini sub network of the Naval Postgraduate School (NPS) computer network. In particular the feasibility of performing various network functions with mini and micro computers is assessed.

**Node Functions and Capacities**

The logical and physical design of a network is aided by mapping functions into network node capabilities, as shown in Figure 2, where the network functions and nodes are obtained from Table 1; an example is given for a distributed network. Node capabilities are defined in advance of this phase of the design. The chart serves to map the desired functional requirements and capacities into a physical implementation. In order to begin the mapping of user requirements to a network configuration, it is first necessary to have decided on an approach or philosophy for network operation, such as a distributed network. Once the general plan for network development has been selected, many of the design choices, concerning such matters as location of text editing services in the network, can be made.

Once node functions and capacities have been assigned and the number of nodes has been identified, a network topology must be selected. The topology is selected on the basis of providing both efficient communication service for the desired node processing functions and high network reliability. Although the use of several connection layers serves to divorce logical processes from physical implementation details and hence insulates the user from the effects of hardware and system software operations, the price paid for this transparency is complex protocols and high overhead.

One attempt to reduce this complexity has been thru the use of bit oriented protocols, which utilize positionally located control fields rather than code set combinations for link control.

Perhaps a more interesting question than topology is the organization and placement of operating system control in a network. The operating system organization exerts as much, if not more, influence over network performance and reliability than does topology. Most prevailing methods of executive control are centralized. All processes share a coherent view of system state. In distributed processing, system state is fragmented into pieces that reside in different processors. Instead of a single system there are several elements that communicate but are locally controlled. The distinguishing characteristic of a distributed operating system is system-wide control of resources without global, centralized state information. Individual processes have only probabilistic knowledge of global system state. This may lead to problems in the control of distributed systems because the totality of local states may be inconsistent with the global state. Reliability improvements can be obtained with a distributed system because the number of bindings-data and process-are reduced. In order to achieve this reduction, control must be decentralized. Many processing nodes, rather than a single node, maintain their own descriptive and state information about their own resources. Thus operating system control is disassociated from any central node or location in the network. This approach will be illustrated later in the section which describes the campus subnet design.

**Link Functions and Capacities**

The second major element in a network which requires mapping from logical functions to physical capabilities is the links which interconnect nodes. This mapping is shown in Figure 2, where, as in the case of nodes, the example is based on the information in Table 1 for a distributed network. Where computer network usage is largely confined to a single location, such as a single campus or corporate location, consideration of network topology is not critical because the geometry of the network is primarily governed by the need to locate terminal nodes in user work areas. There
is not the complex connectivity problem which characterizes the design of nation-wide computer communication networks. The protocols which are used in small networks are also considerably simplified as compared to their large network counterparts. In a large network with many different hosts, operating systems and varieties of communication networks, it is impossible to separate communications from the user and his host, both logically and physically, as, for example, in the ARPANET and other systems. This approach allows user-to-user communication, such as file transfer, to be controlled by user level protocols without having to be concerned with the actual physical flow of data on the communications subnet. In addition, the logical and physical separation of the communications subnet from host activities improves system reliability because the effects of a host or communications processor failure are minimized.

An aspect of network communications that is critical for both single and multiple site networks is the transfer and storage of files. It is critical from the standpoint of the high communication load imposed on the system, as compared to interactive message transfer, and from the standpoint of cataloging, maintaining and synchronizing multitudinous data bases. This is of particular concern in distributed systems where several nodes may catalog and store data bases and where internode file transfer may occur. In a centralized system there is a natural repository for files and their catalog - it is the central node. In a distributed system there is no natural storage location; in fact, the central node's storage is a result of the decentralization of data base management. In a system with distributed data bases there is the problem of whether the cataloging function should be centralized or decentralized. If the former is used, catalog redundancy is eliminated but it is necessary to forward all file access requests and new and deleted catalog entries to one node, thus increasing the communication load. If the latter is employed, it will be necessary to carry duplicate catalog entries, if the user is to be given the flexibility of using, or possibly storing, files at any node. This problem will be addressed in the next section.

Design of Campus Computer Subnet

The above design methodology will now be applied to the design of the micro/mini subnet of the NPS Computer Network. A topological diagram of the proposed system is shown in Figure 3.

Function Assignment for Processing Nodes and Communication Links

In the case of the NPS Computer Network, topology will be determined primarily by the desired location of local nodes to support RJE and interactive terminals. The physical location of processing nodes is immaterial to the user and is not important from a functional design standpoint. Thus, the major design problem is to assign functions and determine capacities for nodes and links.

An objective of the network is to make available to the user a variety of processing and data management capabilities, in terms of computer hardware, software and communications facilities. Two operational modes are envisioned: one in which the user specifies the processing node or service by means of commands from an interactive terminal or by control cards via an RJE; in the second mode the system selects a node based on the type of service requested and distribution of load on the system. The latter mode would probably be the most highly used. In order to implement this type of operation, a front end processor, which we will call the Work Assignment Node (WAN) would be required at each user terminal cluster location for making processing assignments and for initiating the transfer of a job to the assigned node. For simple jobs, the WAN may assign the job to itself.

Several networks have been implemented or proposed in which mini and micro computers serve in one or more of the following capacities: intelligent terminal, communications switcher, job and process scheduler, and I/O interface. A major concern, as discussed earlier, is whether system-wide status information must be available to all nodes or whether each node or cluster of nodes could operate primarily with local status information and in ignorance of much of the remaining network's operation. To adhere to a principle of distributed computing - local autonomy - and at the same time not impose an excessive status information communications load on the network, the following approach would be used:

- Use a tree type network as shown in Figure 3.
- Process a job at the job entry (WAN) node if the node is capable of processing the job and the user has not designated a processing node.
- If the WAN requests permission from a larger node to transmit the job to that node for processing.
- Data retrieval and file updates can be directed to any node from any terminal by means of a WAN.
- Distribute operating system control among nodes, rather than concentrate control at a master node.
- Provide autonomy of operation at each node. This means that each node would be able to function at its own level or a lower level without assistance from a higher level (a micro-based terminal could perform text editing without assistance from a main computer).
- Only involve a node with another node when it is necessary to communicate requests and status information between nodes and to transfer files.
- Provide for logically disconnecting and isolating nodes from the rest of the network for purposes of: (1) reducing system load when network services are not required by these nodes (2) providing experimental,
laboratory or instructional usage without 2. Richard Eckhouse, et al, "An Overview of Two risk of crashing the rest of the system Workshops on Distributed Processing," Computer, and (3) dedicating a node to a particular Vol. 11, No. 1, January 1978, pp. 22-26. function.

Employ mini computer multiplexing and switching of messages and files and work assignment of jobs.

Use a contention type data communication protocol, which will involve a request for transmission only when a node has a need to transmit.25 This mode would be required by the distributed concept, i.e. nodes must initiate communication when the need arises rather than being polled by a central node, and would place a much lighter communication load on the system than would a polling scheme.

Use a decentralized method of data base storage and access. To do otherwise, would defeat the distributed concept. Many small files would be stored in removable cassettes and on floppy discs. For a given processing function, the user would access the files stored under his identification at the processing node. He would have the capability of interrogating the catalog at each node. If a file is not available at the processing node, the user can transfer a copy to the processing node by means of terminal commands. This method leaves open the possibility of multiple versions of a file at different nodes.26 However it appears that for campus computing the user would be expected to be aware of and capable of controlling the condition of his files. He can always request the desired version to be transferred from another node to a processing node.

Some of the network capacities are shown in Figure 3. These were determined by analyzing user requirements. Many additional capacity factors need to be estimated. Space limitations do not permit a discussion of these factors.

Practicality of Mini/Micro Network Functions

One organization that has tried distributed computing recommends a very conservative approach to distributing functions across network nodes.27 This is the experience of a British corporation rather than a university; nevertheless their comments are of interest. They recommend: (1) providing file storage and maintenance centrally and not locally (2) having a central point of control for data base integrity (3) limiting local data handling to spooling (4) controlling software releases from the central site (5) keeping central control of fault reporting. They also report that a distributed system saves on hardware costs and has advantages when the main computer is over loaded.

References

1. Philip H. Enslow, Jr., "What is a Distributed Data Processing System?", Computer, Vol. 11, No. 1, January 1978, pp. 13-21.

2. Richard Eckhouse, et al, "An Overview of Two Workshops on Distributed Processing," Computer, Vol. 11, No. 1, January 1978, pp. 22-26.

3. Hal B. Becker, "Let's Put Information Networks in Perspective," Datamation, March 1978, pp. 81-86.

4. J. H. McFayden, "Systems Network Architecture: An Overview," IBM Systems Journal, Vol. 15, No. 1, 1976, pp. 4-23.

5. R. A. Donnan and J. R. Kersey, "Synchronous Data Link Control: A Perspective," IBM Systems Journal, Vol. 13, No. 2, 1974, pp. 140-162.

6. D. B. McKay and D. P. Karp, "Protocol for a Computer Network," IBM Systems Journal, Vol. 12, No. 1, 1973, pp. 94-105.

7. Ralph G. Berglund, "Comparing Network Architectures," Datamation, February 1978, pp. 79-85.

8. James P. Gray and Charles R. Blair, "IBM's Systems Network Architecture," Datamation, April 1975, pp. 51-56.

9. Gilbert Falk and John M. McQuillan, "Alternatives for Data Network Architectures," Computer, November 1977, pp. 22-29.

10. William Green and Udo W. Pooch, "A Review of Classification Schemes for Computer Communication Networks," Computer, November 1977, Vol. 10, No. 1, pp. 12-21.

11. D. E. Morgan, D. J. Taylor and G. Couteau, "A Survey of Methods for Improving Computer Network Reliability and Availability," Computer, November 1977, pp. 42-50.

12. Richard des Jardins, "Evolutionary Distributed System Design," Proceedings of the Computer Software and Applications Conference, IEEE Computer Society, November 1977, pp. 765-771.

13. Bit Oriented Protocols, Auerbach Publishers, Inc., 53-01-10, 1977.

14. E. Douglas Jensen, "The Honeywell Experimental Distributed Processor-An Overview," Computer, Vol. 11, No. 1, January 1978, pp. 28-38.

15. Richard Eckhouse, et al, "An Overview of Two Workshops on Distributed Processing," Computer, Vol. 11, No. 1, January 1978, pp. 22-26.

16. Harry C. Forsdick, et al, "Operating Systems for Computer Networks," Computer, Vol. 11, No. 1, January 1978, pp. 48-57.

17. A. G. Fraser, "A Virtual Channel System, Datamation, February 1975, pp. 51-56.

18. Peter J. Denning, "Operating Systems Principles for Data Flow Networks," Computer, July 1978, pp. 86-96.

511
19. Fred J. Maryanski, "A Survey of Developments in Distributed Data Management Systems," Computer, February 1978, pp. 28-38.

21. Robert A. Pierce and David H. Moore, "Network Operating System Functions and Microprocessor Front-Ends," Digest of Papers, COMPCON, Spring 1977, pp. 325-328.

23. J. D. Grimes, "Distributed Processing Concepts Using Microprocessors," COMPCON, Spring 1977, pp. 140-144.

25. B. W. Stutzman, "Data Communication Control Procedures," ACM Computing Surveys, Vol. 4, No. 4, December 1972, pp. 197-220.

26. Richard Peck and Eric Manning, "System Architecture for Distributed Data Management," Computer, Vol. 11, No. 1, January 1978, pp. 40-47.

27. Alan J. Shepard, "A British Example of Distributed Computing," Datamation, March 1978, pp. 87-91.
### Application Function Requirements

- **Batch Execution**
  - Degree of Multiprogramming (for given job characteristics)
- **Interactive Execution**
  - Number of Terminal Users (for given virtual memory size allocated to user)
- **File Transfer**
  - File transfer rate

### Application Support Function Requirements

- **Batch Compilation**
  - Compilation Rate (for single program)
- **Interactive Compilation**
  - Compilation Rate (for single program)

### Network Administrative Function Requirements

- **Memory Management**
  - Virtual Address Space
  - Message Concentration - Input to output concentration ratio

### PHYSICAL CAPABILITIES

- **Maxi Processors**
  - Degree of Multiprogramming
  - Batch Compilation Rate
- **Mini Processors**
  - Number of Terminal Users
  - File Transfer Rate
  - Interactive Compilation Rate
  - Virtual Address Space
- **Communications Processor**
  - Message Concentration Ratio
- **Communications Links**
  - File Transfer Rate

---

**Figure 2**. Mapping of Network Functions to Physical Capabilities (Partial List)

**Figure 3**. Proposed Campus Computer Network

513