NASA/CR—2000-209783

Architectural Methodology Report

Chris Dhas
Computer Networks and Software, Springfield, Virginia

January 2000

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized data bases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/CR—2000-209783

Architectural Methodology Report

Chris Dhas
Computer Networks and Software, Springfield, Virginia

Prepared under Contract NAS3–99165, Task 1

National Aeronautics and
Space Administration

Glenn Research Center

January 2000

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A04

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100
Price Code: A04

Table of Contents

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

NASA's Glenn Research Center (GRC) defines and develops advanced technology for high priority national needs in communications technologies for application to aeronautics and space. GRC tasked Computer Networks & Software Inc. (CNS) to describe the methodologies used in developing a protocol architecture for an in-space Internet node. The node would support NASA's four mission areas:

- Earth Science
- Space Science
- Human Exploration and Development of Space (HEDS)
- Aerospace Technology

This report presents the methodology for developing the protocol architecture. The methodology addresses the architecture for a computer communications environment. It does not address an analog voice architecture.

In a computer environment, communications protocols provide for the orderly exchange of information between end systems. The end systems exchange information over a communications path. The physical arrangement may correspond to a computer and a terminal or to two computers that are exchanging information.

A protocol is a logical abstraction of the physical process of communications. Establishing and maintaining communication between end systems calls for a high degree of cooperation. The same set of functions must exist in both end systems. Communications is achieved by having the corresponding or peer layers in the end systems exchange information with one another. The peer layers communicate by means of formatted blocks of data using a well defined set of rules, which are called "protocols". Some functions of protocols are to establish:

- A standard data abstraction(syntax): concerns the format of the data blocks.
- Necessary conventions (semantics): includes control information for coordination and error control.
- A standard communications path.
- Timing: includes speed matching and sequencing.

The establishment of conventions between two communicating entities in the end systems is essential for communications. Examples of the kind of decisions that need to be made in establishing a protocol convention include the nature of the data representation, the format and the speed of the date representation over the communications path, and the sequence of control messages (if any) which are sent.

One of the main functions of a protocol is to establish a standard path between the communicating entities. This is necessary to create a virtual communications medium with certain desirable characteristics. In essence, it is the function of the protocol to transform the characteristics of the physical communications environment into a more useful virtual communications model.

What is the nature of the virtual communications path established by a protocol? Several possible characteristics can be specified. For instance, the addressing structure over the communications path may allow for communications with only one other entity or with several others. Communications may proceed at one level of priority or at several. Messages flowing over the communications path may be sequenced or not. Errors occurring in the data stream may be detected. The control of traffic flow over the communications path may be complex or relatively simple or even nonexistent. Finally there may be various conventions for initiating communications over the path and for terminating it.

The final function of a protocol is to establish standard data elements for communications over the path; that is, the protocol serves to create a virtual data element for exchange. For instance, two computing elements may wish to exchange a stream of characters. Alternatively, they may wish to deal in messages or records or files that correspond to elements of a file system. Other systems may be constructed in which the transferred element is a program or a job. And finally, there are special purpose applications in which the element to be transferred may be a complex structure such as all or part of a graphic display.

Some of the most basic concepts of a communications protocol can be noted here. One is the concept of handshaking. The term handshaking was coined to describe the controlled transfer of data across an interface. One side shakes hands with the other by a sequence of interlocking steps for the purpose of transferring one unit of information.

A second fundamental concept is that of protocol standards. The use of a standard protocol provides a flexibility that can drastically reduce system development time and maintenance effort plus allow adaptation and evolution as requirements change. Suppose "N" different types of information sources have to communicate with "M" types of information receivers. Without a standard protocol, 'N x M' ad hoc protocols are required to support all of the possible connections. The protocols in this case are all special purpose. The preferred method of achieving this interconnection is to introduce the idea of a standard protocol necessitating only 'N+M' implementations

Another important idea is the distinction between protocols and interfaces. Protocol refers to a set of rules for communication between similar processes. In contrast, interface refers to a set of rules for communication between dissimilar processes.

2. PROTOCOL ARCHITECTURE METHODOLOGY

Protocol architecture methodology is a subset of the general methodology used in the design of computer communications systems. Designing a computer communications system is a complex process because of the number of choices available to the designer. The complexity is magnified as each of the design choices is interrelated to several others. Therefore, any methodology that is developed is going to be, at least in part, an artificial structuring of a complex iterative process.

Because of the options available to the designer, an unique protocol architecture methodology does not exist. Figure 1 shows the steps that will be used in this document as part of the protocol architecture methodology.

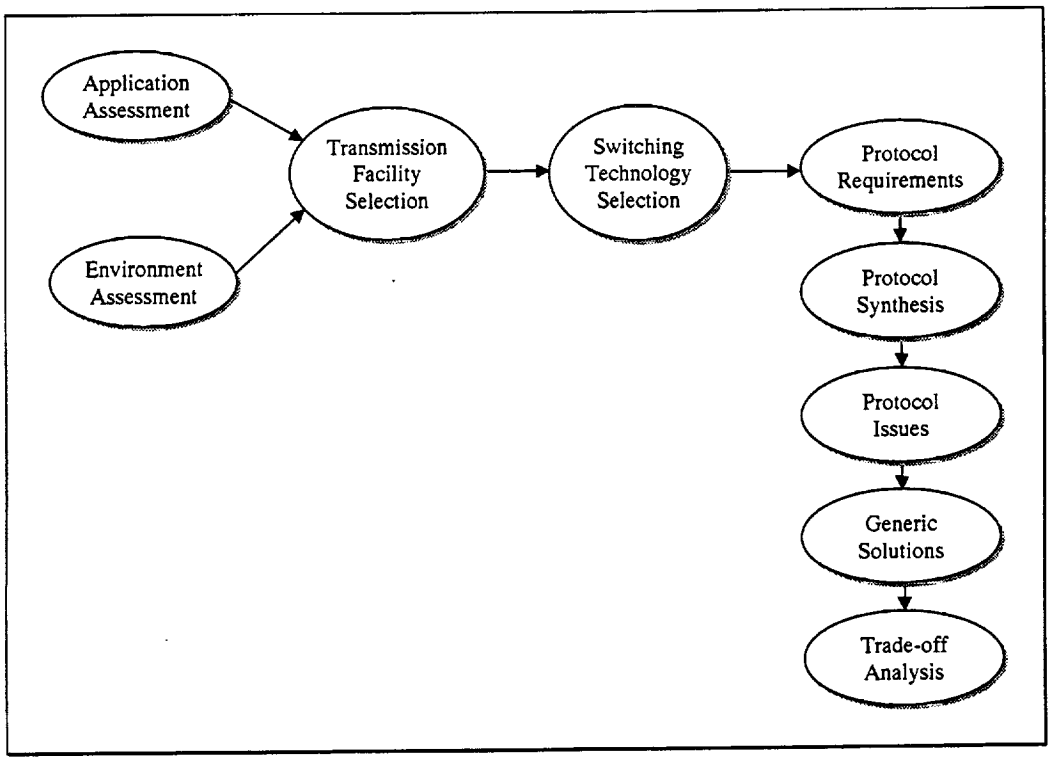

Figure 1. Protocol Architecture Methodology Steps

The reason for existence of the computer communications system is to support end user applications. Therefore, the obvious place to begin employing the methodology is with the overall application assessment step. This step identifies the application requirements that need to be supported by the protocol. A component of the application assessment is the development of the design objective. These include a number of specific performance, functional, and application objectives.

The environmental assessment step is addressed next. The communications environment can be heterogeneous or homogenous. Heterogeneous is a mixed media environment. One needs to take this into account in designing or selecting a protocol architecture. An architecture that will perform well in a homogenous environment may not be suitable for a heterogeneous one. In addition, the need to accommodate existing hardware, software, and protocols will impact the protocol selection process.

The transmission facility selection process uses the results of the application and environmental assessments. The selection of transmission media for the system involves the analysis of the physical transmission paths that could be used to support communications. It may also involve choices among types of uses of the media and methods of obtaining communications service.

The selection of the switching technology is closely related to the choice of the transmission media. This choice centers on the logical communications methods to be employed as opposed to the physical communications paths. The switching method is also closely related to the techniques used for sharing the transmission paths in the network.

Protocol requirements analysis is the central part of the protocol architecture methodology. In this step, the protocol functions needed to support each of the supported applications are identified. The protocol function set that is used corresponds closely to those associated with the International Organization for Standardization (ISO) seven layer Open Systems Interconnection (OSI) protocol architecture reference model. In addition to the application, presentation, and session layer functions, those associated with both connection oriented and connectionless transport, network, and link layers are considered. As the analysis progresses, the functions related to a particular layer are grouped together and those that are not required are eliminated. Then, a determination is made as to whether functions should be moved to different layers of the architecture. In the end, various layers can be consolidated or in some cases removed from the architecture.

Protocol synthesis is the process of developing a conceptual architecture using the results of the protocol requirements analysis step. Protocol synthesis considers the techniques available to implement a protocol function and selects the one that best meets the requirement of the applications it supports. This may involve trade-off and performance analyses. The end result of this process is the conceptual protocol architecture.

The Protocol Issues and Generic Solution to Certain Issues section covers a number of common protocol issues that are well known in the protocol design field. In addition, it also presents generic solutions that have been used in the design of some existing protocols.

3. APPLICATION ASSESSMENT

A key issue in developing practical computer communications protocols is a clear understanding of the basic design objectives for the system in which the protocols will operate. It is obvious that the protocol designer must begin by understanding the objectives of the system designer.

While this may seem elementary, it is not unusual to find large computer communications systems in which the protocols been optimized for certain functions or performance levels which were not originally contemplated in the overall system. These functions can be in conflict with other elements of the communications complex.

The reasons for building computer communications systems are as varied as there are systems in existence, but certain types of rationale recur. These are:

- Performance levels
- Functions provided
- Applications supported
- Cost effectiveness

There may be organizational motivations for developing the system including coordination among different branches of the organization, centralization of operations, or decentralization of operations. It may be advantageous to promote data and program sharing among various groups. There may be financial reasons such as reducing communications or computer costs or facilitating new cost-saving or sales-increasing applications. In addition, there may be technical motivations such as the need for faster turnaround, higher availability, or more growth potential. There are many other reasons why computer communications systems of various sizes and structures have been built and their objectives vary correspondingly. But it is possible to identify three major objectives:

- Performance levels that are required to meet the basic purpose of the computer communications system
- Functions which the system must provide to satisfy its end users
- Applications which the system must support if it is to be successful

3.1. Performance Level

There are a number of performance objectives that can be identified in the design of computer communications systems. They fall into two major categories. The first is speed of communication, meaning the speed with which traffic flows through the system. The second is service characteristics, meaning the grade of service provided to the end user.

3.1.1. Speed of Communications

Speed of communications consists of delay and throughput. Delay is usually measured in average response time and throughput is usually measured by the peak traffic level supported. Development of a protocol architecture requires a through understanding of the parameters that make up delay and throughput and their effect on the protocol architecture.

Delay

The term delay is defined as the time between transmission and delivery of the first bit of the message. There are several components of delay in a communications system. These include:

- Speed of light delay, which is a function of the length of the transmission circuit.
- Transmission delay, which is proportional to the size of the message and inversely proportional to the transmission bandwidth.
- Processing delay incurred at any switching node or store-and-forward facility.
- Queuing delay, which is a function of system load.

The important concept to understand about communications delay is that certain types of traffic require a short delay period in order to achieve a satisfactory response time. These applications include: interactive applications which are communicating with remote computers; database applications making use of query/response; and various real-time applications in which the data must be received a short time after it is generated.

Throughput

System throughput is the number of bits sent divided by the time between transmission of the first bit and delivery of the last bit. Throughput is the effective traffic rate of the system in bits per second. There are a number of components that determine the over-all throughput for a particular system, including the effective end system processing bandwidth and the effective communications bandwidth of the transmission media. It is the nature of throughput that it is a rate which is determined by the lowest component rate in the system. That is, throughput is determined by the slowest component in the system. That component acts as the communications bottleneck.

Just as low delay is required for certain applications, so too high throughput levels are required to meet certain communications needs. Applications such as those for transferring files of data from point to point and those performing real-time communications functions require high throughput. In these cases large amounts of data must be transferred from end system to end system. It is the time required transmitting all the bits which is important rather than the time required transmitting and receiving the first bit.

It is important to realize that delay and throughput are not the same. This is sometimes a confusing point since the use of faster transmission circuits does improve both delay and

throughput. However, the two measures are independent. A simple examination of two cases shows the importance of this difference. First of all, consider a voice grade circuit with a delay of 0.01 seconds and a throughput of 2.4 Kbps. The time to send 100 bits is 0.05 seconds and the time to send 1 million bits is 417 seconds. Now consider a second case in which the transmission is over a satellite circuit with a delay of 0.25 seconds and a throughput of 1500 Kbps. Here the time to send 100 bits increases slightly to 0.25 seconds, whereas the time to send 1 million bits is much less at only 0.92 seconds.

Delay Bandwidth Product

It is also useful to talk about the multiplicative product of these two metrics, often called the delay bandwidth product. Intuitively, a channel between a pair of processes can be thought as a hollow pipe, where the latency corresponds to the length of the pipe and the bandwidth gives the diameter of the pipe. The delay bandwidth product gives the volume of the pipe; i.e., the number of bits it holds. Said another way, if latency (measured in time) corresponds to the length of the pipe, then given the width of each bit (also measured in time), you can calculate how many bits fit in the pipe. For example, a transcontinental channel with a one-way latency of 50 ms and a bandwidth of 45 Mbps is able to hold (sec bits/sec = bits) approximately 280 KB of data.

The delay bandwidth product is important when constructing high performance networks because it corresponds to how many bits the sender must transmit before the first bit arrives at the receiver. If the sender is expecting the receiver to signal that bits are starting to arrive and it takes another channel latency period for this signal to propagate back to the sender, then the sender can send up to two delay bandwidth's worth of data before hearing from the receiver that all is well. (The channel's round trip time rather than just its one-way latency is the second channel latency period.) Also, if the receiver says "stop sending", it might receive that much data before the sender manages to respond.

3.1.2. Communication Service Characteristics

Communications service characteristics consist of availability, data integrity, message integrity, and security. Availability is measured as the percentage of up time. Data integrity is measured as the bit error rate in the system. Message integrity is the measure of the rate of message loss. Security is the rate of message misdelivery.

These characteristics play an important role in the design of a communication protocol architecture and in the design of the communications system itself. To meet a high grade of service, the protocol architecture will include a high level of overhead. The overhead functions are needed to support the grade of service desired.

The implementation of some grade of service functions may be optional. This would allow each of the applications to selectively pick the service characteristics that are required.

3.1.3. Performance Trade-Off

The significance of the difference between delay and throughput is that it creates a situation in which the designer is faced with tradeoffs in determining the performance of the system. One can visualize the situation as a triangle as shown in Figure 2. The three points are in the triangle are low delay, high throughput, and good service characteristics. The triangle defines a set of operating points. The closer the point is to one of the vertices of the triangle the better it satisfies that requirement. It is inherently very difficult to satisfy all three requirements. On an even simpler level, one can consider low delay and high throughput to be requirements for more speed; the opposing goal is to provide better or higher quality services.

For a communications system with a fixed physical capacity and layout to provide low delay, the system must operate with relatively short messages. The messages must be transmitted quickly and with relatively short queues for all transmission facilities so that queuing delay is very small. Likewise, it is desirable to have as few control messages as possible before the data message can be sent.

High throughput on the other hand requires long messages so that the overhead per message can be minimized. High throughput normally requires long queues so that fluctuations in traffic load can be averaged over time and good utilization can be achieved on an average basis. In addition to relatively small amounts of overhead for each message, high throughput requires small amounts of control information.

The provision of good service in terms of data integrity, availability, and security usually requires a number of specific network protocols and functions. Such facilities often require extra overhead on each message, extra processing overhead and specific control messages to perform explicit functions. As one can see, each of these goals is fundamentally in conflict with the others and any system represents a compromise among all three.

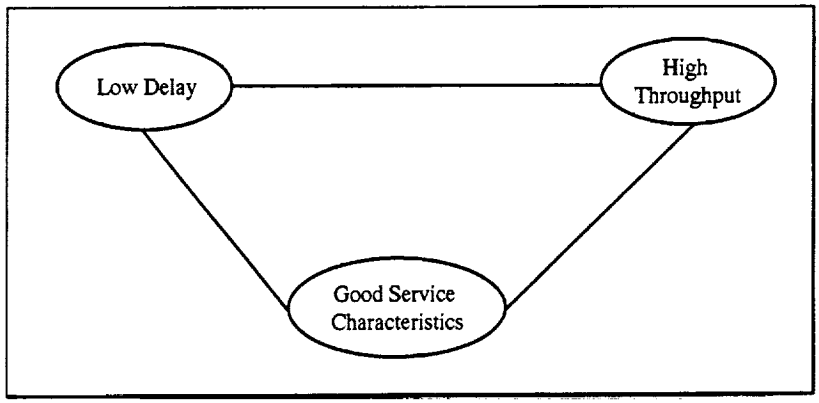

Figure 2. Performance Tradeoffs

Table 1 contains the typical communications characteristics associated with various performance levels. The process of developing a protocol architecture involves trading off these parameters in developing protocol messages as well as control functions.

Table 1. Performance Tradeoffs

Performance	Message Size	Queue Size	Control Messages	Overhead
Low Delay	Short	Short	Few	Low
High Throughput	Long	Long	Few	Low
Good Service	Short	Short	More	Medium

3.2. Functions to be Provided

The next choice in determining the design objectives for a computer communications system is to decide among various functions which have to be provided in each of several areas. There are three main categories of choices - data representation, data transfer, and message related service characteristics.

3.2.1. Data Representation

There are a number of basic design decisions which can be made regarding data representation. For instance, should the communications system support only one data type or code (e.g., ASCII) or should it support the communication of unrestricted data in binary messages? A related decision corresponds to the format of the messages to be exchanged. Should they have a fixed length and fixed fields? This may be appropriate for applications such as voice, whereas other applications may require a free format message in which data can be exchanged in an unrestricted manner. Finally, there are questions regarding the content of the messages.

- Should the communications system deliver these messages in a completely transparent manner or should it provide a certain amount of data and code conversion?

- Should the system perform a significant amount of interpretation of the content of the messages? This may be appropriate in certain kinds of electronic mail systems or in specialized applications.

3.2.2. Data Transfer

The next set of considerations deals with the transfer of the data. Here the issues are basic ones such as error control over the communications path, addressing, priority, and security. There are a number of approaches to address error control. The four basic approaches are:

- Little or no error control
- Error detection by means of check bits
- Relatively complete error control including retransmission of erred messages from the source in the system
- Forward error correction

Other choices include the nature and structure of the addressing in the network and whether it is possible to address messages to more than one destination and to destinations outside of the communication system itself. Another question is the nature of the priority structure to be supported. Some communications systems permit multiple levels of priority and some include preemptive priority for the highest level.

A final consideration is the extent to which secure communications should be supported. This may simply mean the designation of certain groups of addresses as being capable of communicating with each other or it may result in the formal definition or a number of levels of security. It may even result in the decision to transmit part or all of each message in encrypted form.

3.2.3. Message Related Service Characteristics

The final set of decisions concerning system functions involves the message related service characteristics which the system may provide. For instance, what grade of service should the system offer? Some communication systems have been designed such that all traffic competes on a first-come, first-served basis. Other systems permit the regulated flow of traffic through the use of fairness mechanisms. Still others guarantee a certain level of service to users who satisfy specific requirements such as paying enough money, having enough authority, and so on.

There are additional functions that a communications system can provide apart from simply delivering messages to the destination. It may be desirable to notify the source that the message was received. It may be useful to provide the "camp on" service in which the system continues to attempt to deliver a message when the destination is first discovered to be unavailable. Finally, it may be useful to provide notification to the sender of a message when a failure in the system has been detected.

Alternatively, the system may take it upon itself to deliver all messages even in the event of failure. This may call for redundant equipment in the system and elaborate procedures for failure recovery. Other possible services include assistance with message preparation, message storage, retrieval, and statistics on traffic levels and patterns.

3.3. Applications to be Supported

The preceding provides the context for performing an analysis of the applications that are to be supported by the protocol architecture. The next step is to decide which applications are to be supported and to identify their characteristics. A suggested set of characteristics with subdivisions follows:

- Message Size. Size of the data that is transferred over the data communication link.
  - Small. The total message (information exchange data) is between 1 and 1,000 bytes per event.
  - Medium. The message is between 1,001 and 50,000 bytes
  - Large. The message is greater than 50,001 bytes.
- Event Timing. This provides an indication of the occurrence of the data to be communicated. The parameter is defined by:
  - Asynchronous. The event requiring the transmission of data is based upon a demand which occurs at random intervals.
  - Synchronous. The event requiring the transmission data occurs at fixed intervals. This is coupled to precise clock synchronization.
- Persistence. This parameter describes, in general terms, the load presented to the data transfer mechanism. Thus, it indicates the attention level that must be given to the data by the transfer mechanism. The persistence (loading) is described by:
  - Infrequent. The need to communicate data is seldom.
  - Medium. The need to transfer data is less than frequent but more than infrequent.
  - Frequent. It is expected that the need to communicate data is regular but does contain periods of no activity.
  - Continuous. The need to transfer data is continuous without any periods of inactivity.
- Response Time. This is the time between the transmission of information and the receipt of a response that action has been taken or the receipt of the information requested by the original transmission. For example, response time could be grouped into four levels:
  - Level 1 = 1 - 3 Seconds
  - Level 2 = 1 - 3 Minutes

- Level 3 = 10 - 20 Minutes

- Level 4 = Greater than one hour

- Bandwidth Requirement. The estimated channel bandwidth in terms of bits per second that would accommodate the information transfer.

- Precedence. In a mixed application environment where the functions share resources (e.g., the same channel), this is the requirement to give priority to a given data transfer over that of another function.

- Integrity. The integrity level indicates the need for error free data transfer. This is also an estimate of the trust placed in the accuracy of the received data. In general, a rating of high would be equal to an undetected error rate of less than 1 error in 10 gigabits of data. Low would indicate 1 undetected error or less in 100 kilobits. Medium would be an error rate between high and low.

- Availability. The availability is the percent of time that the communication channel must correctly operate and be "available" per unit of time. Thus, an availability of 99.999 means that in one year the channel would be "unavailable" (down) for less than six minutes.

- Security. Security is the requirement to protect the privacy of the data and reveal it only to authorized people or processes. The requirement also includes the need to prevent being denied the receipt of the data to due covert actions or interference of others. The requirement is stated by defining the type of security measures expected to taken to secure the data transmission. This is defined by:

  A = Authentication techniques which provide for confirmation that the true sender and receiver are connected. These may be passwords or certificate style (e.g., PKI) mechanisms.

  E = Data Encryption is required to prevent disclosure of the "plain" data content.

  C = The taking of active countermeasures may be necessary in other to prevent denial of service (e.g., by the use of anti-jamming encoding schemes or redundant channels).

  NR = Not Required. The need for secure communications is not considered necessary for the application type.

- Scalability. This factor describes the predicated growth in the required communication to support the application type. The description is in terms of the multiplier (or increase) of bandwidth and in the rate of increase. In addition, an estimate should be made of the time

period over which the growth will occur. Since space communications require long life times, it is expected that the capacity to allow for growth will be a factor in the overall design.

In broad terms, the scalability parameter is an indication of the growth reserve that would be built into the initial communications architecture.

3.4. Cost Effectiveness

The final consideration in completing the design objectives for a computer communications system is to establish the requirement for cost effectiveness of the system. Cost effectiveness is expressed in terms of how this requirement will be stated and also the specific numerical level of performance, if any.

There are a number of simple measures of the cost-effectiveness of any system. Does the system pay for itself? Does the system cost less than the system it replaces? In the case of a computer communications system, some specific measures might include the cost per month for each subscriber connection or the cost to send a message. Alternatively, one can look at growth measures of the system such as the cost for each additional subscriber, additional message sent, new application, or new geographical location. In other words, when investigating the cost-effectiveness of the system, one must ask the right question, which may be with regard to relative, absolute, or incremental cost. The significance of establishing this fact early in the design process is that the designer of the communications protocols can then attempt to optimize the cost-effectiveness of the protocols in an appropriate way.

The protocol architecture is designed to support end user applications. Sometimes, it is difficult to come up with an architecture that will meet all the application requirements and still meet the cost constraints. Parameters such as delay, throughput, and quality of service push the protocol architecture in opposite directions. At best, the protocol architecture is a compromise among a number of competing parameters.

A summary of application considerations is shown in Table 2.

Table 2. Application Considerations

Factor	Subfactor
Delay	• Speed of Light • Transmission • Processing • Queuing

Factor	Subfactor
Throughput	• Effective End System Processing Bandwidth • Effective Communications Bandwidth
Data Representation	• Data Types • Message Format • Transparent Communications • Data and/or Message Conversion • Message Interpretation
Data Transfer	• Error Control • Addressing • Priority • Security
Message Related Services	• Grade of Service • Message Receipt Notification • Camp on Service • System Failure Notification • Message Preparation • Message Storage • Message Retrieval • Traffic Level and Pattern Statistics
Supported Applications	• Message size • Event timing • Persistence • Response time • Bandwidth requirements • Precedence • Integrity • Availability • Security • Scalability • Load Factor
Delay Bandwidth Product	
Delay/Throughput/Service Trade-offs	
Cost Effectiveness	

4. ENVIRONMENTAL ASSESSMENT

The environment in which the applications will communicate is an important factor in the design of the protocol architecture. The functions to be provided by the protocol architecture depend on whether the environment is homogenous and heterogeneous. Therefore, the first consideration in the choice of a transmission media for data communications is the simple question: "Should a single medium be used or is there an advantage to using a mixed-media system?"

Building the data communications facility with a single medium (homogeneous environment) is certainly the simplest approach and is a good choice when the traffic is anticipated to be uniform. However, a homogenous environment has some drawbacks. It may be a poor match for certain traffic types flowing over the system, may be vulnerable to various kinds of errors or damage, and has the disadvantage that it may be inflexible when the present traffic levels grow or the nature of the traffic changes. A mixed media system is potentially suitable for more traffic types and is certainly more suitable for heterogeneous traffic and for applications in which growth and change are anticipated. In the future, it is anticipated that more and more data communications facilities will employ two or more transmission media to accomplish a variety of system goals.

The next step is to identify the existing hardware and it characteristics. This process becomes simple if legacy systems do not need to be supported. It is rare that a protocol designer has the luxury of starting with a blank slate. From the hardware point of view, the designer has to identify the existing physical interfaces and decide how to provide support for these interfaces in the new protocol architecture environment.

Once the hardware interfaces of the legacy systems are identified, the next step is to identify and understand the existing protocols that must be supported. It may happen that some of the protocols may not be able to work with the new protocol architecture. Or, there may not be a reason to continue supporting these protocols.

A summary of environmental considerations is shown in Table 3.

5. TRANSMISSION FACILITY SELECTION

An important design decision in the construction of a computer communications system of any kind is that of the transmission facilities. The bandwidth required to support the peak requirement of various applications and the inherent error capabilities affect the capabilities to be supported by the protocol architecture. One fundamental choice is the frequency range at which the transmissions will operate. The higher the frequency the more bandwidth in the electromagnetic spectrum is available for communications. It is common to refer to the bandwidth of a communications circuit in terms of its transmission rate in bits per second.

Table 3. Environmental Considerations

Factor	Subfactor
Environment	• Homogeneous • Heterogeneous
Transmission Media	• Single Medium • Mixed-Media Medium
Existing Hardware	
Existing Protocols	
Existing Interfaces	

Determining the bandwidth needed to support peak requirements involves the use of a peak hour traffic estimate. The peak hour is the busiest hour in the year in terms of traffic to be supported by the system. This involves identifying the worst month of the year, the worst week of the month, the worst day of the week, and then the worst hour of the day. The bandwidth needed to support the traffic in that hour is the peak hour bandwidth. A rule of thumb for estimating the peak hour traffic is to use 15 % of the daily traffic.

Starting with an idea of what is required for transmission bandwidth, the next step is to select the transmission medium or media which will be used to carry the traffic. Following this choice one is faced with a variety of questions regarding the type of facility to use. For instance, should the circuits be dedicated or switched, analog or digital, and so on? These choices affect the functions to be supported by the protocol architecture. If, for example, the inherent error rate of the physical transmission facility is 10^{-5} and the application expects an error rate of 10^{-9}, the protocol architecture has to provide additional functions that will enhance the error performance of the link.

5.1. Transmission Media

There is a wide variety of transmission media available for data communications. The most common facility is the standard common carrier circuit. This is generally a voice grade telecommunications circuit. These circuits are available throughout the world on a leased basis, either private or switched lines which can be maintained and reconfigured by the carrier without much difficulty. They can carry traffic over long distances at a variety of speeds. The carrier may use various multiplexing techniques to accommodate a number of such circuits on the same physical line cable or radio frequency.

An alternative to the carrier facilities is to use a pair of wires between two points. This type of facility can he inexpensive and can be used at moderate speeds and distances with modems. However, without a modem the use of a pair of wires is restricted either to slow speeds or to short distances.

Another approach is to use coaxial cable, which is composed of a conductive cylinder with wire in the center. The space between the cylinder and the wire is filled with insulation. Coaxial cable can transmit at a much higher frequency than a pair of wires. Therefore, it can support multi-megabit transmission and is relatively immune to noise. It is inexpensive relative to its speed and can support point-to-point operation and broadcast operation. Finally, one of the advantages of using a coaxial cable system is that it may be possible to interface with existing cable systems such as CATV.

The use of CATV cables for data communications is just emerging. However, it promises to be an effective means of transmission for some systems, since it combines the advantages of coaxial cable with the ease of capitalizing on installed systems for video transmission.

Another form of transmission for data communications systems is optical fiber. The use of optical fiber as a substitute for copper in cable is taking place at a much faster pace in the long haul portion of the network. It has a number of very attractive properties. First of all it is possible to transmit at extremely high bandwidths over optical fiber paths. The cable has very small diameter and weight. Since transmission is by pulses of light, there are no possibilities for electrical crosstalk or interference. This results in very low error rates. It is also projected that optical fiber will ultimately become a low-cost medium.

One transmission medium which has emerged from experimental status to wide-scale operational use is satellite transmission. Communication satellites orbit the earth in a geo-static position functioning as relay stations for earthbound microwave communications links. This is ideal for long-distance communications since the high altitude of the satellite avoids various types of earth-level interference. The satellite communications systems presently in use offer very high bandwidth at low incremental cost per bit per second. One of the other important advantages of satellite communication is that it is inherently broadcast in nature; that is, one sender can communicate with a number of receivers. This has obvious advantages for certain types of applications. The advent of LEO/MEO satellite systems along with inter satellite communication capability makes satellite an attractive transmission medium for a number of applications.

Still another possibility is the use of radio frequency transmission. Long-wave radio systems have been used for many years in telecommunications. More recently, microwave transmission using the high end of the radio frequency range has entered use by common carriers, especially on long-haul communications links. It requires less-frequent re-amplification and no wires. This permits very high bandwidth communication, although microwave transmission requires a line-of-sight path. Therefore, relay towers are usually spaced about 30 miles apart. Another use of radio communication is for mobile data communications on a broadcast basis. This is most appropriate for use in access to larger communications systems in which mobility is essential. Still another application for radio is the connection of large numbers of terminals within urban areas to communications systems. Here the high bandwidth and broadcast nature of the transmission medium make it possible to connect a large number of users at relatively low cost.

In addition to the physical media described so far, there is another possibility for acquiring transmission facilities. That is to lease such facilities from a value-added carrier who provides network services in addition to transmission facilities. These networks are characterized by broad geographical availability and by flexible leased service. Charges for such networks can be based on usage only and can be independent of distance. Also, these charges may include maintenance and upgrading by the carrier. These facilities represent an attractive alternative for users who do not wish to manage their own communications facilities and who may have highly time-varying requirements or requirements which are spread over a wide geographical area. In short they represent a good alternative for users who may not be able to afford a complete communications system of their own.

5.2. Transmission Facility Choices

The next step is to consider the choices that can be made concerning the transmission media. First of all, there is a basic choice as to whether the facilities are public or private. Public transmission facilities are those that can be obtained from the regulated common carrier. In contrast, private transmission facilities are those that the data communications user builds or acquires.

Public data communications has a number of advantages. It is widely available over long distances with small costs to install the system. Service can be leased and maintenance can be obtained from the carrier. Systems can be reconfigured quite easily. On the other hand the offerings are limited to those which the carriers see fit to provide.

Private facilities are often useful for short distances when inexpensive media such as wire or cable can be used. The user must install such equipment and in many cases must purchase and maintain it. This means that the user has a greater degree of commitment to the equipment. On the other hand, the user is free to make use of it in a variety of ways.

Traditionally, analog communications facilities originally designed for voice transmissions have been used for data communications. This is primarily a matter of economics. Not until the last several years have digital facilities been available. Analog circuits allow data communications at a wide variety of speeds over different types of circuits. However, they require modems. Analog circuits are the most widely available offerings and are as common as telephone service.

Digital transmission on the other hand requires no modem and is somewhat more cost-effective. Its primary advantage lies in its enhanced reliability. Since signals are regenerated at each repeater rather than amplified as in the case of analog facilities, no errors are introduced at that level. For this reasons and because digital multiplexing allows more channels to be derived from a given bandwidth than with analog transmission, it is the direction of future growth within the telecommunications industry.

Another basic choice is between dedicated and shared facilities. Dedicated facilities are often referred to as private lines or leased lines and differ from shared facilities, often known as

switched circuits. The user of a dedicated line has the sole use of a pre-established circuit although the physical facilities used for maintaining that circuit may be part of the switched network. The reasons for using dedicated facilities are to achieve improved performance by eliminating contention among users requiring low delay and to have simple communications procedures. The drawbacks to the use of dedicated facilities include higher cost per user than with other approaches and the inherent unreliability of single dedicated communications lines. To avoid total blackout, users of dedicated facilities often use dial-up as a fallback arrangement.

Dedicated digital lines are the most popular types of data communications facilities today. One of the main advantages is that the user has a wide range of facilities from which to choose. However, shared facilities offer the opportunity to reduce the cost per user. Such facilities may be better for traffic in which the peak is much higher than the average. On the other hand, the bandwidth that a user receives is dependent on the load, and contention may result in the user being denied service altogether. Also, more equipment is needed to share the transmission media.

Sharing can be accomplished in a number of ways. The traditional method is by using the switched telephone network. A call from the subscriber reaches the local switching node or central office, is routed to a regional central office, and then to the local central office of the destination. Other possibilities for sharing include multiplexing user traffic onto a single transmission facility with the use of private equipment. Alternatively, the medium to be shared may be broadcast in nature. Sharing can be accomplished implicitly by permitting many transmitters and many receivers on the same communications medium.

Another fundamental distinction is between point-to-point transmission facilities and multipoint facilities. Point-to-point communications is the simplest and the most common. There is no waiting for other subscribers and there are no elaborate signaling procedures. On the other hand, it may become very expensive for many users who need to intercommunicate or even users who need to communicate with a single central location.

Multipoint communication provides reduced cost in some instances, although more complex control mechanisms need to be instituted. These may include polling from a central location or connections establishment, and termination conventions. In any case, there is a certain amount of queuing due to the other subscribers sharing the facility.

A summary of transmission facility considerations is shown in Table 4.

Table 4. Transmission Facility Considerations

Factor	Subfactor
Transmission Frequency	
Analog Circuits	• Widely Available • Variety of Speeds • Modems Required
Digital Circuits	• Less Prone to Errors than Analog • More Cost Effective than Analog • More Channels per Given Bandwidth
Dedicated Facility	• Sole Use of Pre-established Circuits • No Contention Among Users • Simple Communications Procedures • Higher Cost per User • Single Point of Failure • Availability
Shared Facility	• Less Costly • Support for Peak Traffic • User Bandwidth Dependent upon Load • Subject to Contention • More Equipment Needed than for Dedicated Facilities
Public Data Facilities	• Low Installation Cost • Lease from Value-added Carrier • Easily Reconfigured • Carrier Controls Service Offerings
Private Data Facilities	• Acquisition Cost • Higher Maintenance Cost • Control Service Offerings
Facility Cost	• Operation • Maintenance • Upgrade

Factor	Subfactor
Point-to-Point Communications	• Simplest and Most Common • No Waiting for Other Subscribers • Elaborate Signaling Procedures Not Needed • Fully Meshed Network Expensive
Multipoint Communications	• Lower Cost • More Complex Control Mechanisms Needed
Physical Media	• Satellite • Optical Fiber • Coaxial Cable • CATV Cables • Long-wave Radio • Wire Pair

6. SWITCHING TECHNOLOGY SELECTION

The next step in the methodology is the choice of how the various users of the system should share the transmission media. The concept of resource sharing is fundamental to any computer communications system. There are several possible ways of sharing the computer and communications resources. If point-to-point communications media are used, the system can be organized around multiplexing or switching. In the case of multipoint or broadcast communications systems, polling and contention are two alternatives for resource sharing.

6.1. Multiplexing Technologies

Multiplexing is the method by which several channels of communication are combined into one. Communication can be thought of as existing in two domains, a bandwidth domain and a time domain. There are two kinds of multiplexing. One termed frequency division multiplexing (FDM) that allocates a particular section of bandwidth to each channel for all time. The other is known as time division multiplexing (TDM) and assigns time slots to channels in which each slot occupies the whole bandwidth. FDM is commonly used in telephone systems in which many individual telephone signals are carried on high-bandwidth circuits. FDM can also be used for digital transmission. TDM, on the other hand, has been used in some small telephone exchanges but has been used more often in computer communications systems for digital TDM.

Time Division Multiplexing

The two basic problems in digital TDM are how to ensure that the frames are identified correctly at the receiver and how to deal with an incoming signal if its clock is not exactly correct for the

multiplex channel. The common solution to these problems is to carry framing information by some kind of coding in the data. TDM systems can implement either bit-multiplexing or byte multiplexing schemes. Such multiplexers can be organized into hierarchical tree networks in which slow speed multiplexers feed into higher speeds multiplexers. Such systems are often called synchronous networks.

Statistical Time Division Multiplexing

A third method for performing multiplexing is with dynamically addressed blocks. This is sometimes known as statistical TDM or concentration. In this case, blocks of data are all of the same size but do not occupy an assigned fixed place in the frame. This technique is called concentration since the sum of the input rates can exceed the output. Obviously, if traffic is entering on all of the input lines at a peak rate, then congestion will result. The advantages of statistical TDM are clear. It offers the opportunity for increased use of the channel and decreased delay for each individual user.

6.2. Point-to-Point Subnetwork Technologies

There are three possibilities for switching technologies that can be used to construct communications networks. They are circuit (or line) switching, message switching, and packet switching.

6.3. Circuit Mode Technology

A circuit-switching network provides service by setting up the dedicated physical path between two communicating subscribers. The complete circuit is set up by a special signaling message. The signaling message passes all the way through the network and a return signal informs the source that data transmission may begin. The path setup time is usually on the order of seconds. The entire path remains allocated to the transmission until the source releases the circuit.

Circuit switching is the common method used for telephone systems. It is an appropriate method for communication when the two subscribers have identical equipment such as voice telephone instruments and no speed or code matching is necessary. Also it is appropriate if the users communicate at a fairly constant rate for a long period of time. Circuit switching is inefficient for bursty traffic, which has a high peak-to-average ratio.

6.4. Message Mode Technology

In message switching only one channel is used at a time for a given transmission. The message is a logical data unit and it first travels from the source node to the next node in the path. Each node in the path stores the message on disk and forwards the message to the next node. This process repeats in a store-and-forward manner with queuing delays at any node where the selected

channel is busy. Such systems have been built to optimize the use of network lines and to remove the burden of communications responsibility from the user. Speed and code conversion can be performed in such networks. Examples of message-switching networks include the Telex network, the AUTODIN I network, and the SITA airline system. Delays in the message-switching systems built in the last decade are usually on the order of many minutes.

6.5. Packet Mode Technology

The final switching technology is packet switching. It is similar to message switching except that secondary storage is not used in the network. Messages are split into smaller units called packets, which are routed independently on a store-and-forward basis through the network. Thus, many packets of the same message may be in transmission simultaneously. This is one of the main advantages of packet switching. Packet switching offers the most dynamic switching technology since it makes effective use of circuit bandwidth with sophisticated switches.

A simplified delay comparison can be made among circuit, message, and packet switching. Circuit switching includes a connection delay at the switch followed by the transmission of a setup signal. The cycle is repeated at each line until the return signal is generated at the destination. Then the data is sent from source to destination with relatively little delay. For message switching there is a small setup delay and then the entire message is transmitted from the source to the first node in the network. After this node fully receives the message, the sequence is repeated through the network. For packet switching the message is broken up into smaller packets and transmission is started. After the first packet is received at the second node, it can be relayed while the first node is beginning the transmission of the next packet. It can readily be shown that packet switching is the technology, which offers the least delay, and that message switching is an improvement over circuit switching in most cases. It is worthwhile to note, that all of these comparisons are dependent on technology and that there is nothing implicit in the functions performed by these switching methods which makes one superior to the other. That is, a message switching system with high-speed lines can outperform a slow-speed packet-switching system.

Packet mode switching technology allows dynamic sharing of a given bandwidth. Hence, it is not a big surprise to find that there is more than one way to accomplish the goal of sharing. Some of these techniques are closely associated with the state of the technology and the environment. There are various packet mode methods such as Ethernet, token ring, FDDI, and ATM in the local area environment. Technologies such as X.25, Frame Relay, SMDS, and ATM seem to fit into the metropolitan and wide area network environments. The trick is to select the technology that best supports the application requirements. Table 5 presents a comparison of the switching technologies

Table 5. Comparison of Switching Techniques

Characteristic	Circuit Switching	Message Switching	Packet Switching
Physical Connection	Yes	No	No
Real Time	Yes	No	Yes
Data Storage	No	Yes	Temporary
Blocking with Overload	Yes	No	Yes
Delays with Overhead	No	Yes	Yes
Error Control	No	Yes	Partial
Speed/Code Conversion	No	Yes	Yes
Delayed Delivery	No	Yes	Possible
Multiaddress	No	Yes	Possible

6.6. Multipoint (Broadcast) Techniques

The next set of techniques are those that are appropriate for multipoint communications systems in which a given source can transmit to more than one receiver, or a given number of sources can all transmit to a central receiver, or both. Such systems are characterized by partially or completely broadcast transmission media. There are two basic methods for performing multipoint communications. The first is polling from a central point; the second is contention-based communications.

Polling has been widely implemented for inquiry/response systems as well as in multidrop networks in which a single telephone circuit is shared by more than one subscriber. In all cases a controller is used to initiate and carry out polling. There are two types of polling disciplines in common use. In the roll call or bus polling system, each message source is interrogated in turn by the central source. On the arrival of a polling message, the source polled transmits all waiting messages to the central source. On completion of the message transmission, the next source is polled. In the hub or distributed polling discipline, the central source initiates polling by interrogating the message source at the end of the loop. This source transmits its waiting data and then signals the next source in line to begin transmitting. At the completion of the cycle with all sources connected into the loop interrogated, the central source regains control.

The second major method for connecting a number of users over a multipoint transmission medium is contention. In a contention system each subscriber attempts to communicate information to a central source at the time the information becomes available or at a locally determined time soon thereafter, as opposed to waiting until interrogated by the central source.

A summary of switching technology considerations is shown in Table 6.

Table 6. Switching Technology Considerations

Factor	Subfactor
Multiplexing	• FDM • TDM
TDM	• Bit Multiplexing • Byte Multiplexing • Synchronous Network
Statistical TDM	• Dynamically Addressed Blocks • Congestion • Increased Channel Usage • Decreased Delay
Circuit Switching	• Dedicated Physical Path • Path Setup Time • Constant Rate Traffic for Long Duration • Inefficient for Bursty Traffic
Message Switching	• Optimize Use of Network Lines • Code Conversion • Queuing Delays
Packet Switching	• Dynamic Sharing of Bandwidth • Dynamic Switching • Secondary Storage Not Needed
Multipoint	• Partial or Complete Broadcast Transmission • Polling from Central Point • Contention-Based

7. PROTOCOL REQUIREMENTS ANALYSIS METHODOLOGY

The applications assessment step identified the end-to-end application related requirements. This step identifies the protocol requirements needed to support each application. Note that some the requirements identified in the applications assessment step influence indirectly the protocol requirements as well. Figure 3 presents the steps involved in the protocol requirement analysis process.

We start with identifying the protocol requirements for each applications. The OSI reference model for open system communication has codified the protocol layers required in an end system and how the peer protocols communicate using the interfaces. Our process uses the OSI

reference model and its associated functions as a guide. The functions and services associated with the layers in the OSI reference model are presented in detail in section 8.6 through 8.12.

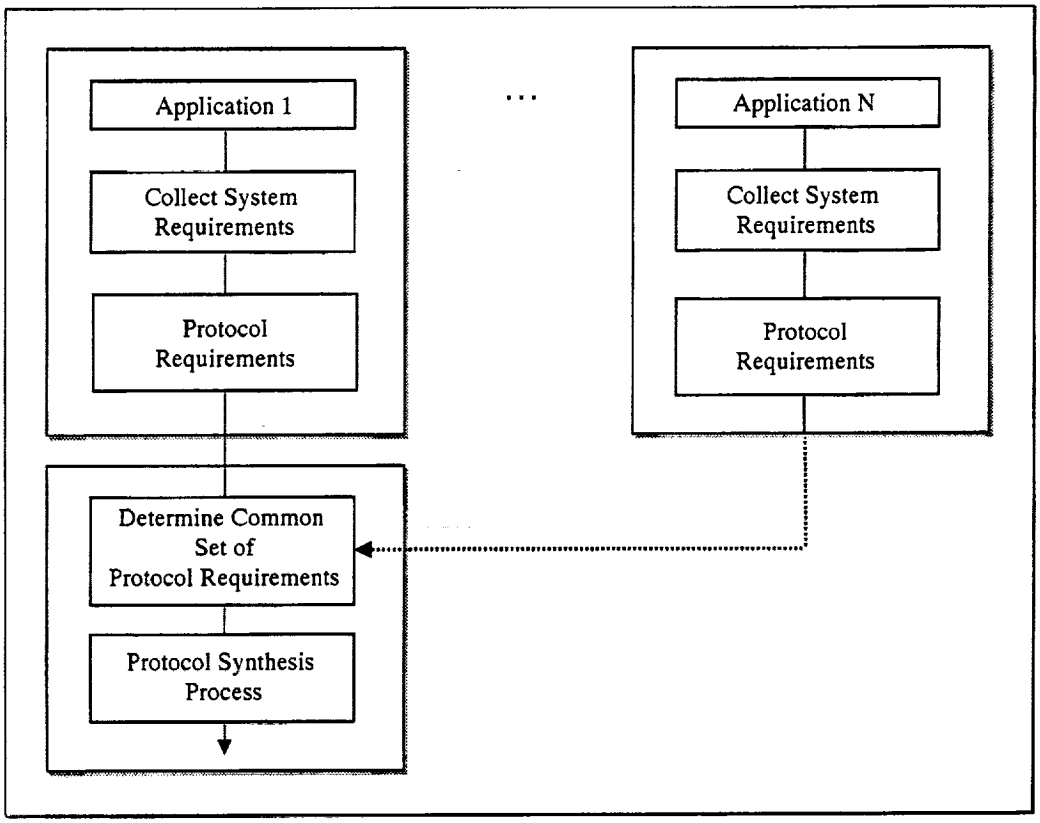

Figure 3. Protocol Requirements Analysis Process

Table 7 shows the protocol functions that an application may need. The characteristics column presents the functions associated with the OSI reference model. Note that during the analysis phase, we have included both connectionless and connection-oriented services at various layers of the model. This ensures we take into account all of the protocol requirements.

Once the protocol requirements for all the applications has been captured, the next step is to group the protocol functions and eliminate those functions that are not required. Then, a determination is made as to whether functions should be moved to different layers of the architecture. In the end, various layers can be consolidated or in some cases removed from the architecture. The consolidated set of protocol requirements form the input to the protocol synthesis process.

Table 7. Protocol Requirements Analysis

Characteristics	Application 1	Application 2	Application N
Application Layer Functions			
Applications identifications			
Application context required			
Presentation context required			
Presentation service requirements			
Session service requirements			
Turn management			
Synchronization			
Recovery from temporary loss of presentation connection			
Synchronous operation			
Asynchronous operation			
Real time			
User Authentication			
Linked operation			
Presentation Layer Functions			
Presentation connection establishment, and its termination with its peer			
Negotiation of syntax for presenting application data			
Syntax transformation including data compression			
Data transfer by using the services of the lower layers			
Dialogue control			
Session Layer Functions			
Connection establishment			
Connection release			
Connection abort			
Activity management			
Major and minor synchronization			
Symmetric synchronization			
Resynchronization			
Exception			

Characteristics	Application 1	Application 2	Application N
Negotiated release			
Abort services			
Data transfer by using the services of the lower layers			
Transport Layer Functions - Reliable Transport Required			
Precedence			
Segmentation			
Graceful closing			
Flow control			
Integrity			
Duplicate control			
Multiplexing			
Sequence control			
Blocking			
Error detection			
Error recovery			
Expedited data transfer			
Transport Layer Functions - Reliable Transport Not Required			
Segmentation			
Data transfer			
Network Layer Functions - Connection-Oriented			
Connection establishment and its maintenance			
Multiplexing of connections			
Re-initialization or reset of connections			
Addressing, routing and relaying			
Normal and expedited data transfer			
Sequencing and flow control, segmentation and re-assembly			
Error detection, notification and possibly recovery			
Network Layer Functions - Connectionless			
Addressing, routing and transfer of data			
Error detection and notification			
Access control			
Source authentication			

Characteristics	Application 1	Application 2	Application N
Confidentiality			
Possible segmentation			
Multicasting			
Data Link Layer Functions - Connection-Oriented			
Connection establishment and release			
Splitting of data link connections			
Delimiting and synchronization of protocol-data-units			
Error detection and recovery			
Flow control and sequenced delivery			
Data Link Layer Functions - Connectionless			
Data transfer			
Error detection			

8. PROTOCOL SYNTHESIS

8.1. Protocol Architecture

An architecture is defined by the Institute of Electrical and Electronic Engineers as the structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. Architecture is an abstract model of some part of the real world. In the communications environment, an architecture is a model of the organization and behavior of networks consisting of interconnected, communicating computer systems and applications.

An architecture serves two important functions. First, it creates a global conceptual framework within which the relationship among individual components can be studied and explored at a common level of abstraction. This framework encourages broadly based solutions to problems, since it places each component in an abstract context that shows its interaction with other components. Second, it servers as the basis for a formal description of the characteristics of individual components. In addition, it establishes a common reference point for the behavior of the designs and implementations that result from it.

Systems are considered "open" by virtue of their mutual adherence to one or more open systems standards. The standards specify the aspects of the behavior of an open system which are directly relevant to its ability to communicate with other open systems. There are a number of open system specifications and standards, each of which belongs to a particular architecture (e.g., OSI or TCP/IP). An architecture is typically described by a reference model that expresses the

organizing principles of the architecture and provides a framework within which the various services and protocols and the relationship among them may be defined.

The protocol architecture selected should be such that the over all system will provide the requisite performance levels and be cost effective. The protocol design is the key element in successful resource sharing between end systems. One of the important choices in the overall philosophy of protocol design is whether the protocols should be transparent to the user or whether they should perform a virtual communications function. A transparent protocol provides a service of which the user is not aware. A virtual protocol provides a service to the user such as creating a virtual file, but the service is a logical entity only and has no actual physical existence. The question of which approach to take is difficult to answer in general. However, protocols should be transparent whenever possible since this allows for the simplest kinds of operations. On the other hand, they should be virtual when that is necessary to allow higher level protocols to be built on top of lower level protocols. In this case, the lower level protocols must establish certain virtual communications entities.

A related choice is between a single level of protocol verses several layers of protocol arranged in a hierarchy. The advantage of a one level protocol is that it is relatively simple and efficient to implement. Most one level protocols are application-dependent, special-purpose protocols, and they work well in that context. The advantage of having multiple levels of protocols is that they provide a separation of functions, which is always useful in designing a complex system. This permits the segregation of responsibility for resource management into several areas corresponding to the several different resources. Most important, it provides support for evolutionary changes of the protocol since each level of the protocol acts as a separate module in implementation terms.

The recommended approach is for multiple levels of protocol in all cases but levels of protocol which permit special-purpose protocol levels to be substituted or used in parallel with the standard protocols. This approach combines the advantage of complete generality of the multiple levels of protocols with the use of special purpose protocols for special functions.

The concept of layering is good, but how many layers? The OSI reference model specifies seven layers, whereas the Internet architecture uses only five. The fact that the number of layers is different is not deeply significant. What is significant is the way in which OSI and the Internet architectures allocate functions among the layers and their relevance to the support of the distributed applications.

8.2. Protocol Header

A protocol is a well-defined set of rules for conversation among end systems. It consists of a set of rules and formats (semantics and syntax) which determine the communication behavior in the performance of layer functions. Each layer uses a protocol to exchange user data with a peer layer. To convey the rules of the protocol from sender to receiver, protocol control information (header information) is attached to the user data. The header information is meaningful only to

the peer of a given layer. When the combination of user data and header information is passed down to an adjacent, lower layer, it is considered as user data in that layer. As each layer performs its well-defined functions, the process of data encapsulation is repeated until the physical layer is reached.

8.3. Layering Principles

The OSI reference model was developed to support communications among open systems. Zimmermann in a classical article on OSI architecture provides principles that were used in developing the seven layer model.

- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy.
- A layer should be created where a different level of abstraction is needed.
- Each layer should perform a well defined function.
- The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
- The layer boundaries should be chosen to minimize the information flow across the interfaces.

8.4. Layer Design Issues

Some of the key design issues that must be addressed for computer communications system are present in several layers of the architecture. Some of the more important ones are presented below.

Every layer needs a mechanism for identifying senders and receivers. Since a network normally has many end systems (some of which have multiple processes), a means is needed for a process on a system to specify with whom it wants to communicate. As a consequence of having multiple destinations, some form of addressing is needed in order to specify a specific destination.

Another set of design decisions concerns the rules for data transfer. In some systems, data only travels in one direction (simplex communication). In others it can travel in either direction, but not simultaneously (half-duplex communication). In still others it can travel in both directions at once (full duplex communication). The protocol must also determine how many logical channels the connection represents, and what the logical channel priorities are. Many protocol

architectures support at least two logical channels per connection, one for normal data and one for urgent data.

Error control is an important issue because physical communications circuits are not perfect. Many error detecting and error correcting codes are known, but both ends of the connection must agree on which one is being used. In addition, the receiver must have some way of telling the sender which messages have been correctly received and which have not.

Not all communication channels preserve the order of messages sent to them. To deal with a possible loss of sequencing, the protocol must make explicit provisions for the receiver to allow the pieces to be put back together properly. An obvious solution is to number the pieces, but this solution still leaves open the question of what should be done with pieces that arrive out of order.

An issue that occurs at every level is how to keep a fast sender from swamping a slow receiver with data. Various solutions have been proposed. Some of them involve some form of feedback from the receiver to the sender (either directly or indirectly) about the receiver's current situation. Others limit the sender to an agreed upon transmission rate.

Another problem that must be solved at several levels is the inability of all processing to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting, and reassembling messages. A related issue is what to do when processes insist upon transmitting data in units that are so small that sending each one separately is inefficient. Here the solution is to gather together several small messages heading toward a common destination into a single large message and disassemble the larger message at the other end.

When it is inconvenient or expensive to set up a separate connection for each pair of communicating processes, the underlying layer may decide to use the same connection for multiple, unrelated conversations. As long as this multiplexing and demultiplexing is done transparently, it can be used at any layer. Multiplexing is needed in the physical layer, for example, when the traffic for all connections has to be sent over at most a few physical circuits.

When there are multiple paths between source and destination, a route must be chosen. Sometimes this decision must be split over two or more layers. For example when multiple domains are involved, a high level decision might be made based on privacy laws, and a low level decision made to choose one of many available circuits based on the current traffic load.

Another issue is the type of services provided by the layers. Layers can offer two different types of service to the layers above them - connection-oriented and connectionless services. For a connection-oriented network service, the user first establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a pipe. The sender pushes messages in at one end, and the receiver takes them out in the same order at the other end.

In a connectionless service each message carries the full destination address, and is routed through the system independent of the other messages. Normally when two messages are sent to

the same destination, the first one sent will be the first to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first. With a connection oriented service this is impossible.

Each service can be characterized by a quality of service. Some services are reliable in the sense that they never lose data. Usually reliable service is implemented by having the receiver acknowledge the receipt of each message so that the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable.

Reliable connection oriented service has two minor variations: message sequence and byte streams. In the former the message boundaries are preserved. In the latter, the connection is simply a stream of bytes with no message boundaries. A third type of connection oriented service is the unreliable service. The connectionless service can also support minor variations called unreliable datagram, acknowledged datagram, and request-reply.

8.5. OSI Reference Model

This section contains a description of the seven layers of the OSI reference model. The reference model is used as a guide in developing a protocol architecture. The functionality of each layer and its interface to the succeeding/preceding layer has been well defined. A specific protocol stack may choose to define more or less than seven layers. However, the functionality of most protocols stacks defined after the publication of the OSI reference model, as well as of the ones defined prior to its publication, can be mapped to the OSI reference model. Figure 4 shows the OSI architecture reference model.

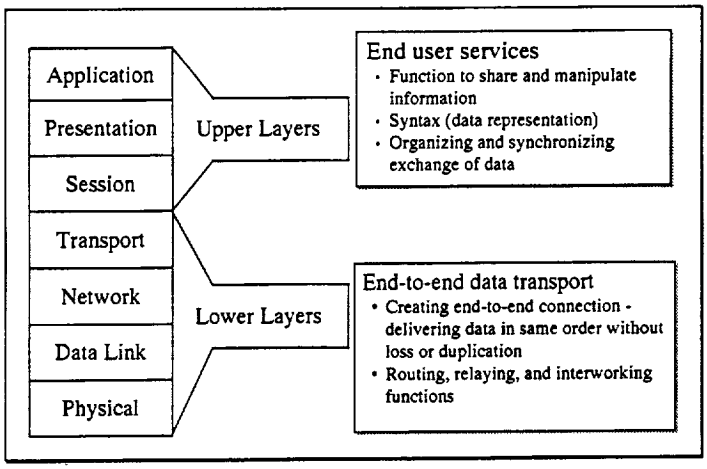

Figure 4. OSI Architecture Reference Model

8.6. Application Layer

The application layer provides access to OSI services. It is the layer in which the distributed applications reside and in which they access the network service. The application layer is the highest layer of the reference model. The application layer is considered the source and sink of all data being exchanged across the network. An application can be thought as consisting of a local component and communications component. The local component provides the end-user interfaces, whereas the communications component consists of the entity that provides the distributed communications capabilities. In OSI the communications component is called the Application Entity (AE) and the sum of the parts that make up a distributed application is the Application Process (AP). The distributed application service selects functions from a large common pool of application service elements (ASEs).

The OSI reference model has identified a number of core capabilities or service elements that provide specific services. These service elements provide:

- The ability to initiate and terminate communication across a network and to ascertain prior to attempting to transmit data that the called application has all the facilities and capabilities required to interpret and operate on the data about to be sent. This capability is provided by the association control service element (ACSE).

- The ability to structure conversations between distributed applications and the ability for an application to recover from disruption of underlying communications services without loss of data. This capability is provided by the reliable transfer service element (RTSE). It provides applications access to dialogue control capabilities such as activity and turn management, synchronization, and resynchronization.

- The ability to perform functions at remote systems is accomplished by the remote operations service element (ROSE).

The general services provided by the application layer are:

- Applications identifications
- Application context required
- Presentation context required
- Present service requirements
- Session service requirements
- Turn management
- Synchronization
- Recovery from temporary loss of presentation connection
- Synchronous operation
- Asynchronous operation
- Real time
- User authentication
- Linked operation

When defining an application layer, one may ask the following questions:

- What common functions will be needed to support a host of applications?
- Are the functions specified by ASEs sufficient to support the application entities being envisioned?
- If not, what additional functionality is required?
- What are the characteristics of the lower layer with which the application layer will interact?
- Are there any off-the-shelf protocols that will satisfy all the requirements?
- If not, is defining, developing and implementing new protocols justified by any marginal gains in functionality?

8.7. Presentation Layer

The task of preserving the semantics of the data exchanged between a sender and receiver across the OSI network is handled by the presentation layer. This layer performs the transformation from the local syntax used by each application entity to a common transfer syntax. The presentation service is presented in terms of the facilities it provides. The connection establishment and connection termination facilities provide presentation connection management between communicating application entities.

In OSI, certain functions reflected by application service elements are really performed in the session layer (token management, synchronization, and checkpointing). The presentation layer does not provide these services directly but instead passes these service primitives between the application and the session layers. Therefore, for applications that require direct manipulation of session services, the presentation layer offers applications pass-through facilities to services offered by the session layer. These pass-through services are collectively called dialogue control. In addition the presentation layer provides four forms of information transfer services: normal, typed, capability, and expedited data.

The presentation layer implements the following functions:

- Connection establishment, and its termination with its peer.
- Negotiation of syntax for presenting application data.
- Syntax transformation including data compression, if needed.

- Data transfer by using the services of the lower layers.
- Dialogue control.

While defining a presentation layer one may ask the following questions:

- Do data encoding and decoding need to be performed in the presentation layer or somewhere else? How does pass-through affect overhead and processing requirements?
- What are various presentation schemes that might be used for data representation between remote entities?
- Is there a need for a separate presentation layer?
- Can the presentation layer function be performed at the application layer?

8.8. Session Layer

Information exchanged between end systems can be viewed as having two fundamental components. The first is the information transfer moving information from source to destination. In general, this is the responsibility of the transport service. The transport service is responsible for the transfer of the unstructured data transparently between end systems. It does not worry about where the application data begins and ends. However, the applications need to know the structure of the data.

The second fundamental component is preserving the structure of the data defined by the application processes. This function belongs to the session service. Therefore, the section layer provides:

- Connection establishment
- Connection release
- Connection abort
- Synchronization
- Resynchronization
- Conversation control
- Activity management
- Exception
- Negotiated release
- Abort services
- Data transfer by using the services of the lower layers

A number of session services are related and are logically grouped into functional units. These are:

- Kernel
- Data transfer
- Activity management
- Major and minor synchronization
- Symmetric synchronization
- Resynchronization
- Exception
- Negotiated release
- Abort services

When defining a session layer, the questions to be asked are:

- Will application entities require multiple sessions active at the same time? If the answer is "no", then most likely the session layer is not needed.

- Can the session layer functionality be moved to another layer?

- Would it be easy for the application layer to provide application specific session services?

8.9. Transport Layer

The transport layer is the basic end-to-end building block of end system networking. Everything above the transport layer is distributed application oriented. Everything below the transport layer is transmission network oriented. The transport layer provides a reliable data pipe for the upper layers as part of a connection-oriented service, and a simple datagram delivery as part of a connectionless service.

During the connection-oriented operation, the data stream submitted to the transport layer by the source transport user must be delivered to the destination transport user without loss. There may be no duplication of any octets in the data stream, and the octets must be delivered in the same order as that in which they were submitted. The transport layer must also provided end-to-end error detection and recovery, which involves the detection of errors introduced into the data stream by the network.

Both the connection-oriented and connectionless modes employed at the transport layer must optimize the use of network resources, given the quality of service objectives specified by the transport user.

The following end-to-end functions are elements of the connection-oriented transport service:

- Precedence
- Segmentation
- Graceful closing

- Flow control
- Integrity
- Multiple routing option
- Duplicate control
- Multiplexing
- Sequence control
- Blocking
- Concatenating
- Error detection
- Error recovery
- Expedited data transfer

The elements of the connectionless transport service are:

- Segmentation
- Data transfer

The transport layer plays a critical role for the end systems. End systems in this context are distinguished from systems that are concerned only with the intermediate functions (routing, relaying, switching, and transmission) of networking. The idea is to provide a genuinely network independent transport service, which can be provided as reliably over connectionless internets as over connection-oriented networks. The transport layer functions are to some extent determined by the quality of service requested by the transport user and the type of network connection available at the time of connection establishment.

8.10. Network Layer

The main function of network layer (and the lower layers) is to provide a data transfer capability across the communications subnetwork. The functionality provided is specific to the subnetwork and must be implemented by the end systems as well as the intermediate systems. The intermediate systems in the subnetwork have the responsibility for routing and relaying information. The network layer shields the transport layer from all concerns about the subnetwork.

The network layer may provide connection-oriented or connectionless data transfer. The functionality of the network layer for connection-oriented transfers involves:

- Connection establishment and its maintenance
- Multiplexing of connections
- Re-initialization or reset of connections
- Addressing, routing, and relaying
- Normal and expedited data transfer
- Sequencing and flow control, segmentation and re-assembly
- Error detection, notification, and possibly recovery

The functionality of the network layer for connectionless data transfer involves:

- Addressing, routing, and transfer of data
- Error detection and notification
- Access control
- Source authentication
- Confidentiality
- Possible segmentation
- Multicasting

The network layer functionality exists not only in end systems but also in the intermediate systems of the subnetwork. The functionality required in the network layer is determined to some extent by the characteristics of the subnetwork. These characteristics include network topology, network error characteristics, and suitability for broadcast. Addressing, routing, and the ability to transfer data are the most important functions of this layer and any specification of this layer must deal with them at a minimum.

8.11. Data Link Layer

The purpose of the Data Link layer is to provide functional and procedural means to establish, maintain, and release connections between network entities and to transfer user data. The task of the data link layer is to convert the raw bit stream offered by the physical layer into a stream of frames for use by the network layer. Various framing methods are used, including character count, character stuffing, and bit stuffing. Data link protocols can provide error control to retransmit damaged or lost frames. To prevent the sending of data from over running a slow receiver, the data link protocol provides flow control. The sliding window mechanism is widely used to integrate error control and flow control in a convenient way. The data link layer offers various service, including:

- Unacknowledged connectionless service
- Acknowledged connectionless service
- Acknowledged connection-oriented service

Unacknowledged connectionless service consists of having the source system send independent frames to the destination system without requiring the destination system to acknowledge them. No connection is established beforehand or released afterwards. If a frame is lost due to noise on the line, no attempt is made to recover it in the data link layer level. This class of service is appropriate when the error rate is very low so recovery of lost data is left to the higher layers.

The next step up in terms of reliability is acknowledged connectionless service. When this service is offered, connections are not used. However, each frame sent is individually acknowledged.

The most sophisticated service the data link layer can provide to the network layer is connection-oriented service. With this service, the source and destination systems establish a connection before any data is transferred. Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received. Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order. The connection-oriented service provides the network layer processes with the equivalent of a reliable bit stream.

The functionality of the data link layer for connection-oriented data transfer involves:

- Connection establishment and release
- Splitting of data link connections
- Delimiting and synchronization of protocol-data-units
- Error detection and recovery
- Flow control and sequenced delivery

The functionality of the data link layer for connectionless data transfer involves:

- Data transfer
- Error detection

To specify a data link protocol, one must ask the following questions:

- Is error detection needed at the data link layer? If error detection is needed, error recovery or error correction may be invoked, or the data unit may simply be discarded.

- Is error recovery needed at the data link layer? If the physical layer is extremely error prone, error recovery will be required to make up for the errors at the physical level. Error recovery mechanisms can add considerable complexity to the data link layer.

- Is error correction needed at the data link layer? Error correction may be recommended for a physical layer that involves excessive propagation delays.

- Is flow control needed? Flow control can also add significant complexity at the data link layer. If alternative congestion management schemes are used by the subnetwork, data link layer flow control may not be needed.

8.12. Physical Layer

The physical layer provides mechanical, electrical, functional, and procedural means to establish, maintain, and release physical connections and for bit transmission over a physical medium. The services provided to the data link entities include physical connection establishment and in-sequence transmission of bits over a data circuit.

In most situations one will not be designing a physical media while defining a protocol stack. Actually, one has to decide from a range of existing and potential physical media where the protocol stack may be supported. Characteristics of each physical media may require functions be provided in the upper subnetwork layers. As noted above, some of the functions in data link layer are mandated by the characteristics of the physical media.

8.13. Protocol Stack Development Steps

The development of a protocol architecture is an evolutionary process rather than a revolutionary one. As mentioned earlier, technology and the operating environment have a great influence on the development a protocol architecture. In general, it is less risky and more cost effective to use existing protocols to satisfy the architecture requirements. If existing protocols do not provide all the functions needed, consideration should be given to modifying one of them before designing and developing a new protocol. The OSI architecture can be used as a guide in allocating functions to different layers in the architecture.

The OSI and the popular TCP/IP protocol architectures have some functions that are similar. However, OSI and TCP/IP have used significantly different approaches in constructing distributed system applications. OSI assumes that distributed applications operate over a strict hierarchy of layers and are constructed from a common tool kit of standardized application service elements.

TCP/IP makes no such assumption, insisting only that distributed applications operate over a common end-to-end transport service. OSI's approach is general and flexible, and its emphasis is on modularity and reuse of common mechanisms. In contrast, TCP/IP's approach is more application specific and may lead to redundant implementation of the same function in many different applications. In most cases this results in greater efficiency and performance, but is more costly.

Given the contrast in styles, the TCP/IP approach to building applications is called a vertical one; i.e., each application is developed independently, top to bottom. The OSI approach (consistent with the notion of layering) is called the horizontal approach. Since the OSI architecture model is more structured and useful in identifying protocol functions, the OSI architecture reference model forms the basis for the protocol development methodology.

The steps in the protocol development methodology are:

- Start with what is given (applications, physical layer)
- Use the OSI seven layer architecture and functional definitions as a guide
- Use the hierarchical model rather than strict layering principles
- Group the required functions into layers

- Match the protocol requirements developed against the OSI layer model
- Generate a conceptual protocol stack
- Validate the layers against other standard and non-standard protocols
- Perform trade-off analysis

9. PROTOCOL ISSUES AND GENERIC SOLUTION TO CERTAIN ISSUES

9.1. Connection Setup and Teardown

The next area to be addressed is how conversations are established and terminated. Usually, this is provided by means of a virtual connection between the sender and receiver. When the connection is established, the sender and receiver exchange various kinds of identification. This includes specification of the parameters for the conversation. It also provides a means whereby the conversation can be disconnected and, more importantly, reconnected in the event of error. These procedures must be able to operate in the face of errors or total failure in the link itself and in the computer equipment at each end.

9.2. Data Transparency

Both user data and control information flow together through a network. Techniques and rules are necessary to distinguish data bits from control bits. Alternatively, one can view this situation as telling where a particular data block begins and ends without restricting the content of the data blocks. Two general methods for achieving data transparency are in widespread use. The first is byte stuffing. It is used in older protocols like BSC. The second is bit stuffing and it is used in HDLC. These protocols work by assigning a control meaning to certain data patterns; i.e., the beginning of a block is flagged with a certain pattern of ones and zeros. In the byte stuffing case, this is a pair of characters. In bit stuffing, it is a series of eight bits. To achieve data transparency (i.e., to be able to send these patterns of characters or bits in a message), the sender adds special bytes or bits which the receiver removes, preserving the original data pattern. Whenever the control byte needs to be transmitted as data, it is doubled. Whenever the receiver sees two control bytes in a row, it removes one of them. The bit stuffing procedure works similarly.

9.3. Failure Recovery

The protocol procedure must recover from complete failures of the system or the circuit. Usually this means that the procedure must allow for explicit resynchronization of various communications parameters. At a fundamental level, this means that the system using the control procedure must continuously perform a test to see that the link is operating normally. When the

link is determined to be inoperative, both ends must take appropriate measures so that they both understand that the circuit is inoperative, even though in physical terms it may be up in one direction. After both ends know that the link is down, they begin to monitor the circuit. After it is working correctly again, they establish a new virtual connection. These mechanisms must be failsoft, so that the link control procedure can continue to operate no matter what type of errors occur in the transmission of data or control information.

9.4. Error Control (Error Detection and Retransmission)

To deliver accurate communications under normal circumstances, it is important that the link control procedure deal with transmission errors. There are many possible methods for error control over a transmission link. The most practical approach is to provide an error detection mechanism for each transmitted message and to allow for retransmission by the source. With this general class of procedures, the retransmission can be triggered either by a negative acknowledgement from the destination when it does not receive the intended message or a scheme of positive acknowledgments from the destination when it does receive messages, or by a timeout mechanism in the source so that it retransmits message when it fails to receive positive acknowledgments. Generally, positive acknowledgement systems are preferred since negative acknowledgements by themselves are not sufficient.

Error detection can be performed by a variety of means. Parity checking is one means. It involves a simple one-bit indicator giving the parity of the binary sum of the data. A more complicated mechanism is cyclic redundancy checksums. It involves a more comprehensive algebraic process that is capable of detecting a large numbers of erred bits.

When a block has been detected to be in error, the destination can choose to ignore the block. After a certain amount of time (usually equal to the expected time to receive a positive acknowledgement), the source can retransmit the block.

In general, it is necessary for the source to identify the blocks that it is transmitting by some kind of number. This number can be as simple as a sequence number which counts from zero to some fixed limit and then recycles. The destination can detect missing blocks by this procedure. It is also sufficient for the detection of duplicate blocks since the destination can maintain the numbers of transmission(s) which it is currently expecting. Any messages with numbers lower than the lowest number expected or higher than the highest number expected are automatically duplicates.

9.5. Sequencing

Sequencing of messages is often employed to insure that traffic leaving is virtually the same as the traffic entering. The usual solution to the problem of keeping data blocks in order is to assign a sequence number to each block to be used for the detection of missing and duplicate blocks and to maintain state information at the sender and receiver. One bit can be used for each block that

is in transmission between sender and receiver. This represents a small addition to the error control logic, which is a reason why many procedures employ sequencing.

9.6. Flow Control

Flow control is concerned about how to match the sender's transmission rate with the receivers ability to accept traffic. A good direct solution is an explicit allocation of resources. That is, the receiver explicitly notifies the sender of its ability to accept traffic. This can be done by allocating a certain amount of storage for messages to be transmitted or by indicating a transmission rate that can be supported. The problem is that the receiver must not become overloaded with traffic it cannot accept. Any system which deals with flow control by allocating a number of messages to the sender are approximating a system of notifying the sender of the receiver's instantaneous reception rate. The concept of a window of a number of messages allowed to be in transit from the source to the destination is one method for approximating this technique.

Most flow control systems are based on some form of feedback control. That is, the sender continues to transmit messages until the receiver begins to cut back on the amount of allocation it allows the sender. Many present day link control procedures do not have very sophisticated flow control processes. The feedback in these systems often consists of a simple on/off control. Therefore, the efficiency of such a link is somewhat less than could be expected with a more finely tuned procedure. Another approach is to use an error control mechanisms for flow control purposes. This gives more flexibility than a simple on/off control. A final type of flow control within a link control procedure is simply to discard messages that cannot be accepted and to rely on some higher level protocol to retransmit them later.

9.7. End-to-End Protocols

There is considerable controversy at the present time over whether a store-and-forward subnetwork of nodes should concern itself with end-to-end transmission procedures. Many feel that the subnetwork should be close to a pure packet carrier with little concern for maintaining message order, for ensuring high levels of correct message delivery, or for message buffering in the subnetwork. Others believe that the subnetwork should take responsibility for many of the end-to-end message-processing procedures. However, many design issues remain constant whether the functions are performed at the host or subnetwork level.

9.7.1. Pipelining and Message Size

Any practical network must allow for multiple messages simultaneously being in transit between the source and the destination to achieve high throughput. If, for example, one message of 2,000 bits is allowed to be outstanding between the source and destination at a time and the normal network transit for the message including destination-to-source acknowledgment is 100

milliseconds, then the throughput rate that can be sustained is 20,000 bits per second. If slow lines, slow responsiveness of the destination host, and great distance cause the normal network transit time to be half a second, then the throughput rate is reduced to only 4,000 bits per second. Similarly, pipelining is essential for most networks to improve delay characteristics; data should travel in reasonably short packets.

Low delay requirements drive message size smaller, network and host lines faster, and network paths shorter (i.e., fewer node-to-node hops). High throughput requirements drive the number of packets in flight up, message overhead down, and the number of alternative paths up. The design process should balance these factors.

9.7.2. Error Control

Source-to-destination error control is comprised of three tasks: detecting bit errors in the delivered messages, detecting missing messages or pieces of messages, and detecting duplicate messages or pieces of messages.

Detecting bit errors in the delivered message is done in a straightforward manner through the use of checksums. A checksum is appended to the message at the source and the checksum is checked at the destination. When the checksum does not match at the destination, the incorrect message is discarded, requiring it be retransmitted from the source. Several points about the manner in which checksumming should be performed are worth of note.

- If possible, the checksum should check the correctness of the resequencing of the messages, which possibly got out of order in the traversal of the network.

- A powerful checksum is more efficient than alternative methods such as replication of a critical control field; it is better to extend the checksum by the number of bits that would have been used in the redundant field.

- Unless encryption is desirable for some other reason, it is simpler (and just as safe) to prevent delivery of a message to an incorrect host through the use of a powerful checksum than it is to use an encryption mechanism.

- Node-to-node checksums do not fulfill the same function as end-to-end checksums because they check only the lines, not the nodes.

Inherent characteristics of packet-switching networks are that some messages or portions of messages (i.e., packets) will fail to be delivered and that there will be some duplicate delivery of messages or portions of messages. Missing messages can be detected at the destination through the use of one state bit for each unit of information which can be simultaneously traversing the network. An interesting detail is that (for the purpose of missing message detection) the state bits used must precisely cycle through all possible states. For example, stamping messages with a time stamp does nothing for detecting missing messages because (unless a message is sent for

every "tick" of the time stamp) there is no way to distinguish the case of a missing message from the case where no messages were sent for a time.

Duplicate messages can be detected with an identifying sequence number such that messages arrive from a prior point in the sequence are recognized as duplicates. Duplicate messages can arrive at the destination quite a long time after the original. The sequence number must not complete a full cycle during this period. For example, if the goal is to be able to transmit 200 minimum-length messages per second from the source to the destination and each needs a unique sequence number and if it is possible for messages to arrive at the destination up to 15 seconds after the initial transmission from the source, then the sequence number must be able to uniquely identify at least 3,000 messages.

It is usually easy to calculate the maximum number of messages that can be sent during some time interval. It is more difficult to limit the maximum time after which duplicate messages will no longer arrive at the destination. One method is to put in each message a timer that is counted down as the message traverses the network. If the timer ever counts out, the message is discarded as too old. This guarantees that no messages older than the initial setting of the timer will be delivered to the destination. Alternatively, one can make a reasonably good approximation of the maximum arrival time through studying all of the worst-case paths through the network and all the worst-case combinations of events which might cause messages to be delayed in the network.

In either case, there must be mechanisms to resynchronize the sequence numbers between the source and the destination at node start-up time and to recover from a node failure. A good practice is to resynchronize the sequence numbers occasionally even though they are not known to be out of step. A frequency with which to do redundant resynchronization would be every time a message has not been sent for longer than the maximum delivery time. In fact, this is the maximum frequency with which the resynchronization can be done (without additional mechanisms).

If duplicates are to be detected reliably, the sequence number at the destination must function without disruption for the maximum delivery time after the "last message" has been sent. If it is desirable or necessary to resynchronize the sequence numbers more often than the maximum time, an additional "use" number must be attached to the sequence number to uniquely identify which "instance" of this set of sequence numbers is in effect. Of course, the packets must also carry the use number.

The next point to make about end-to-end error control is that any message going from source to destination can potentially be missing or duplicated; i.e., not only data messages but also control messages. In fact, the very messages used in error control (e.g., sequence number resynchronization messages) can themselves be missing or duplicated and a proper end-to-end protocol must handle these cases.

Finally, there must be some inquiry-response system from the source to the destination to complete the process of detecting lost messages. When the proper reply or acknowledgement has not been received for too long, the source may inquire whether the destination has received the

message in question. Alternatively, the source may simply retransmit the message in question. In any case, this source inquiry and retransmission system must also function in the face of duplicated or lost inquiries and inquiry response control messages.

As with the internode acknowledgment and retransmission system, the end-to-end acknowledgment and retransmission system must depend on positive acknowledgments from the destination to the source and on explicit inquiries or retransmissions from the source. Negative acknowledgments from the destination to the source are never sufficient (because they might get lost) and are only useful (albeit sometimes very useful) for increased efficiency.

9.8. Storage Allocation and Flow Control

One of the fundamental rules of communications systems is that the source cannot simply send data to the destination without some mechanism for guaranteeing storage for that data. In very primitive systems one can guarantee a rate of disposal of data (as to a line printer) and not exceed that rate at the data source. In more sophisticated systems there seem to be only two alternatives. Either one can explicitly reserve space at the destination for a known amount of data in advance of its transmission. Or, one can declare the transmitted copy of the data expendable, sending additional copies from the source until there is an acknowledgment from the destination. The first alternative is the high-bandwidth solution. When there is no space, only tiny messages travel back and forth between the source and destination for the purpose of reserving destination storage. The second alternative is the low-delay solution; the text of the message propagates as fast as possible.

In either case storage is tied up for an amount of time equal to at least the round trip time. This is a fundamental result – the minimum amount of buffering required by a communications system (either at the source or at the destination) equals the product of round trip time and the channel bandwidth. The only way to circumvent this result is to count on the destination behaving in some predictable fashion (an unrealistic assumption in the general case of autonomous communicating entities).

High throughput and low delay are conflicting goals. From experience and analysis it is known that if both high throughput and low delay are desired, there must be mechanisms to handle each. This is true, in particular, for the storage allocation mechanism. It has occasionally been suggested (mainly for the sake of simplicity) that only the low-delay solution be used; i.e., messages are transmitted from the source without reservation of space at the destination. Those making the choice to never reserve space at the destination frequently assert that high bandwidth will still be possible through use of the following mechanism. The source sends messages toward the destination, notes the arrival of acknowledgments from the destination, uses these acknowledgments to estimate the destination reception rate, and adjusts its transmissions to match that rate. Such schemes may be quite difficult to parameterize for efficient control and, therefore, may result in reduced effective bandwidth and increased effective delay.

The greater the sums of transmit time and response time, the looser and less efficient the feedback loop will be. In fact, there appear to be oscillatory conditions which can occur when performance degrades completely. (If there is much of a possibility of message loss, the acknowledgment and retransmission system should allow selective retransmission of messages rather than requiring a complete window of messages to be retransmitted to effect retransmission of the specific messages requiring it. Otherwise, message retransmission will use excessive bandwidth.)

The above discussion assumes that all mechanisms are attempting to minimize the probability of message discard. The effects of discarding (reduced effective bandwidth and increased effective delay) could drastically reduce performance. This could happen if the communications channel or the destination solve their internal problems (e.g., potential deadlocks) with cavalier discarding of messages.

Further, the above discussion assumed the destination was able to minimize the probability of discard. While this may be possible for a single source, it is unlikely that the destination will be able to resolve (in a way that does not entail excessive discards) the contention for destination storage from multiple uncoordinated sources. Detrimental contention for destination storage, in the absence of a storage reservation mechanism, happens practically continuously under even modest traffic loads and in a way uncoordinated with the rates and strategies of the various sources. As a result, well-behaved hosts (those with sending and receiving rates well matched) may unavoidably be penalized by interference from the excess transmissions of poorly behaved hosts.

In addition to space to hold all the data, there must be space to record what needs to be sent and what has been sent. If a message will result in a response, there must be space to hold the response. Once a response has been sent, the information about what kind of answer was sent must be kept for as long as retransmission of that response may be necessary.

9.8.1. Precedence and Preemption

The first point to note about precedence and preemption is that the total transit time being specified for most new switching networks is on the order of less than a few seconds (often only a fraction of a second). Thus, the traditional specifications (e.g., low-priority traffic must be able to preempt all other traffic so that it can traverse the network in under two minutes) no longer make much sense. When all messages traverse the network in less than a few seconds, there is generally no need to specify that top-priority traffic must preempt other traffic or to specify the relative precedence between the other types of traffic. Priority can be used, however, to admit traffic into the network selectively.

10. ROUTING PROTOCOLS AND ISSUES

It is critical that the routing algorithm be reliable in the face of node and line failures. Indeed, the accurate, reliable operation of the routing algorithm is most important when they occur. Therefore, the momentary or prolonged malfunction of any network component should not interfere with routing, even if the routing program in a node or routing data transmitted on a line is affected. The routing process is intrinsically vulnerable since if one node starts sending out incorrect routing data, the whole network can be affected. It is of paramount importance because if it fails, the network is unusable.

A basic requirement is that the routing algorithm should arrive at a steady-state solution given a static set of input data. The choices it makes should be correct given the input data and should not oscillate. This is an elementary goal, but it should not be overlooked either in the early specification and testing of the program or in its later operation.

The preceding requirement is trivial compared with the requirement that the routing process should adapt to changes in network topology and traffic. This principle influences the design of the routing scheme in several ways. The routing program must be efficient enough to run often in real time as the input change. It must have higher priority than data handling. The network can become badly congested if old routing is used in the face of heavy traffic or when the traffic pattern has shifted greatly, or when a line has gone down. The routing algorithm should adapt as quickly as possible to minimize the effects of suboptimal or infeasible routing. It should adapt smoothly and uniformly over all the nodes affected.

In using the shared resources of the network, it is important that the routing algorithm arrive at globally optimal choices. Routing choices can stabilize at many points in a given situation. In all but the best case, some network resources are being wasted and some network traffic handled inefficiently.

A related and conflicting goal is that the routing algorithm be fair to competition for shared resources. It is easy to construct examples which show that strict global optimality would completely shut off traffic between some nodes. This is unfair and the routing algorithm must have some mechanism to allow such traffic to get through.

10.1. Performance Measures

The routing algorithm should be evaluated in terms of measured performance and cost. The performance of a routing algorithm must be considered in terms of four factors of fundamental importance for the network as a whole.

- Delay. The network topology and the traffic levels of the moment determine the theoretical minimum for delay. The routing program should come as close as possible to that minimum, particularly as concerns interactive traffic.

- Throughput. The situation here is parallel. There is a maximum throughput level for a given network topology and traffic mix. There may well be a tradeoff between good

performance for delay and throughput. The high throughput rates are most important for bulk traffic.

- Cost. The routing algorithm can affect the cost of network utilization by its demands for line bandwidth, node bandwidth, and node storage. Routing over as few lines and nodes as possible and choosing underutilized lines and nodes can keep bandwidth demands down. Keeping network queues short can reduce storage requirements.

- Reliability. The routing process is also related to the reliability of network connectivity. The reachability decision should always be correct. To the extent that it is sometimes wrong or slow in adapting, the network is less reliable.

10.2. Cost Measures

Five specific costs are likely to be incurred by any routing plan. Their effects should be balanced.

- Nodal bandwidth. The recalculation of best routes represents a demand for CPU processing time at the nodes, which is an overhead factor reducing the CPU capability for data processing. Routing should have a higher priority than message processing so this reduction comes off the top.

- Nodal delay. While the routing computation is proceeding with high priority, data flow through the node is delayed representing a direct cost as well.

- Nodal storage. The routing algorithm needs storage for routing information and for the program itself. There may have to be input and output buffers for routing messages. All these represent storage not available to the main message buffer pool and therefore another overhead factor.

- Line bandwidth. The fractional utilization of the line bandwidth for routing depends on the size of the routing messages, their frequency, and the bandwidth of the circuit itself. This line bandwidth reduction also comes off the top because of the priority nature of the messages. In reducing the effective rate of the line, it raises the virtual cost.

- Line delay. The final cost factor to consider is line delays due to routing messages. These delays increase linearly with routing message frequency, quadratically with message length, and quadratically with decreasing line bandwidth.

10.3. Control Alternatives

One fundamental step in designing a routing algorithm is the choice of the control regime to be used in the operation of the algorithm. Nonadaptive or fixed algorithms make no real attempt to adjust to changing network conditions. The nodes exchange no routing information, and no

observations or measurements are made at individual nodes. Centralized adaptive algorithms use a central authority which dictates the routing decisions to the individual nodes in response to network changes. Isolated adaptive algorithms operate independently with each node making exclusive use of local data to adapt to changing conditions. Distributed adaptive algorithms use internode cooperation and the exchange of information to arrive at routing decisions.

11. TRADE-OFF ANALYSIS

Once the protocol functions have been allocated to layers, the protocol architecture will take form. There are many functions to choose from as shown in the protocol analysis form above. Inherent in the protocols that incorporate the functions are parameters; e.g., window size and line speed. The values selected for the parameters will affect the performance of the architecture. Once the parameters have been identified, a trade-off analysis should be preformed to identify the values that will provide the best performance for the situation.

A desktop analysis of the interaction between the parameters is usually not practical. Analytical modeling techniques (such as the use of queuing theory models) are needed to assess the effect of the interactions. A model can be used to assess the impact of:

- Delay as a function of window size
- Delay as a function of line speed
- Throughput as a function of window size
- Throughput as a function of line speed
- Line error rate as a function of header size
- Line error rate as a function of message size

The sensitivity of performance to changes in parametric values can be analyzed and the best set of values chosen for the architecture.

12. PROTOCOL COSTING CONSIDERATIONS

Direct cost trade-offs are made in the design of the specific protocols that comprise the architecture. Prior to that point, decisions will be made that impact the life cycle cost of the architecture. Specifically, the decisions made in identifying protocol functions that are needed and the allocation of them to specific layers in the protocol architecture are life cycle cost drivers. Existing protocols are significantly less expensive to implement that newly designed ones. If the designer allocates functions in such a way that new protocol development is minimized, the life cycle cost will also be minimized.

The use of a methodology that focuses on open systems protocols such as the OSI model for architecture development should result in a lower life cycle cost solution. The designer will be making cost impacting decisions in the process of allocating the protocol functions. If as

suggested earlier the designer considers the existing generic solutions in the allocation process, the chances are greater that more use will be made of existing protocols.

13. REFERENCES

1. ATM Forum PNNI Working Group, af-pnni-0055.000, Private Network-Network Interface Specification Version 1.0, Mar 1996.

2. Bhasin, K. B., et al., Enhancing End-to-End Performance of Information Services Over Ka-Band Global Satellite Networks, NASA/TM-97-206297, Dec 1997.

3. Charalambous, P. C. et al., Performance Evaluation of TCP Extensions on ATM over High Bandwidth Delay Product Networks, IEEE Communications Magazine, Jul 1999.

4. Chitre, P. and Yegenoglu, F., Next-Generation Satellite Networks: Architecture and Implementations, IEEE Communications Magazine, Mar 1999.

5. Comer, D. E., Internetworking with TCP/IP, Vol. I: Principles, Protocols and Architecture, Prentice Hall, 1991.

6. Cuevas, E. G., The Development of Performance and Availability Standards for Satellite ATM Networks, IEEE Communications Magazine, Jul 1999.

7. Durst, R. C., et al., TCP extensions for Space Communications, Proc. ACM Mobicomm'97, Nov 1996.

8. Ghani, N., and Dixit, S., TCP/IP Enhancements for Satellite Networks, IEEE Communications Magazine, Jul 1999.

9. Goyal, R. et al., Traffic Management for TCP/IP over Satellite ATM Networks, IEEE Communications Magazine, Mar 1999.

10. Hoe, J. C., Improving the Start-up Behavior of a Congestion Control Scheme for TCP, Proc. ACM SIGCOMM'97, Aug 1996.

11. Information Technology - Open Systems Interconnection—Basic Reference Model: The Basic Model. International Standard, ISO/IEC 7498-1, 1994.

12. Internet Engineering Task Force, Internet Protocol, (Postel, J. B., Ed.), Request for Comments (RFC) 791, Sep 1981.

13. Internet Engineering Task Force, Postel, J. B.; and Reynolds, J. K., File Transfer Protocol (FTP), Request for Comments (RFC) 959, (Postel, J. B., and Reynolds, J. K.; Ed.), Oct 1985.

14. Internet Engineering Task Force, TCP Extensions for High Performance, RFC 1323. (Jacobson, V.; Braden; R, Borman,.. D.; Ed.), May 1992.

15. Ivancic, W. D., et al., NASA's Broadband Satellite Networking Research, IEEE Communications Magazine, Jul 1999.

16. Keshav, S., A Control-Theoretic Approach to Flow Control. Proc. ACM SIGCOMM'91, Sep 1991.

17. Liu, Z., et al., Evaluation of TCP Vegas: Emulation and Experiments, Proc. ACM SIGCOMM'95, Aug 1995.

18. Mathis, M., et al., TCP Selective Acknowledgement Options, RFC 2018, Oct 1996.

19. Mathis, M., Mahdavi, J., Forward Acknowledgement: Refining TCP Congestion Control, Proc. ACM SIGCOMM'96, Aug 1996.

20. Mertzanis, I. et al., Protocol Architecture for Satellite ATM Broadband Networks, IEEE Communications Magazine, Mar 1999.

21. Partridge, C., and Shepard, T. J., TCP/IP Performance Over Satellite Links, IEEE Network, Sep/Oct 1997.

22. Zhang, Y., et al., Satellite Communications in the Global Internet-Issues, Pitfalls, and Potential, Proc. INET'97, 1997.

APPENDIX A. ACRONYMS

ACSE	Association Control Service Element
AE	Application Entity
AP	Application Process
ASE	Application Service Element
ATM	Asynchronous Transfer Mode
AUTODIN	Automatic Digital Network
BSC	Binary Synchronous Communication
CATV	Cable Television
CPU	Central Processing Unit
FDDI	Fiber Distributed Data Interface
FDM	Frequency Division Multiplexing
HDLC	High Level Data Link Control
HEDS	Human Exploration and Development of Space
ISO	International Organization for Standardization
KB	Thousand bits
Kbps	Thousand bits per second
LEO	Low Earth Orbit
Mbps	Million bits per second
MEO	Middle Earth Orbiting
OSI	Open Systems Interconnection
PKI	Public Key Infrastructure
ROSE	Remote Operations Service Element
RTSE	Reliable Transfer Service Element
SMDS	Switched Multimegabit Data Service
TCP/IP	Transport Control Protocol/Internet Protocol
TDM	Time Division Multiplexing

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 2000	Final Contractor Report

4. TITLE AND SUBTITLE

Architectural Methodology Report

5. FUNDING NUMBERS

WU-632-50-51-00
NAS3-99165
Task 1

6. AUTHOR(S)

Chris Dhas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Computer Networks and Software, Inc.
7405 Alban Station Ct.
Springfield, Virginia 22150

8. PERFORMING ORGANIZATION REPORT NUMBER

E-12068

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—2000-209783

11. SUPPLEMENTARY NOTES

Project Manager, Thomas Wallett, Communications Technology Division, NASA Glenn Research Center, organization code 5610, (216) 433-3673.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category: 32 Distribution: Nonstandard

This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The establishment of conventions between two communicating entities in the end systems is essential for communications. Examples of the kind of decisions that need to be made in establishing a protocol convention include the nature of the data representation, the format and the speed of the date representation over the communications path, and the sequence of control messages (if any) which are sent. One of the main functions of a protocol is to establish a standard path between the communicating entities. This is necessary to create a virtual communications medium with certain desirable characteristics. In essence, it is the function of the protocol to transform the characteristics of the physical communications environment into a more useful virtual communications model. The final function of a protocol is to establish standard data elements for communications over the path; that is, the protocol serves to create a virtual data element for exchange. Other systems may be constructed in which the transferred element is a program or a job. Finally, there are special purpose applications in which the element to be transferred may be a complex structure such as all or part of a graphic display. NASA's Glenn Research Center (GRC) defines and develops advanced technology for high priority national needs in communications technologies for application to aeronautics and space. GRC tasked Computer Networks and Software Inc. (CNS) to describe the methodologies used in developing a protocol architecture for an in-space Internet node. The node would support NASA's four mission areas: Earth Science; Space Science; Human Exploration and Development of Space (HEDS); Aerospace Technology. This report presents the methodology for developing the protocol architecture. The methodology addresses the architecture for a computer communications environment. It does not address an analog voice architecture.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Network; Architecture	61
	16. PRICE CODE: A04

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	