AN OPTIMIZATION ANALYSIS
OF FRAME ARCHITECTURE IN SELECTED PROTOCOLS

by

Sham Chakravorty

Project Report submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

Dr. Fred J. Ricci, Director

Dr. Daniel J. Schaefer

Dr. Richard Nieporent

March 1993

Blacksburg, Virginia

C.2

AN OPTIMIZATION ANALYSIS OF FRAME ARCHITECTURE IN SELECTED PROTOCOLS

by

Sham Chakravorty

Committee Chairman: Dr. Fred J. Ricci

Electrical Engineering

(ABSTRACT)

In the current multi-protocol networking environment where a large number of networks coexist and provide transmission, switching and network management services, multiple protocol conversions take place between the networks. These conversions occur in switches, fileservers and hosts, and are essential for maintaining smooth data flow. Two typical activities takes place during a translation, first, conversion of one type of data frame to another, and second, clock synchronization. These activities consume time and affect the throughput of the overall system. In this paper, the merits and demerits of each of a select group of protocols are discussed, especially relative to their frame architectures and overhead characteristics, with an eye toward optimizing some of the protocols.

The analytical approach chosen for this paper comprises a detailed review and analysis of the characteristics of specific protocol framing architectures in all the layers of Open Systems Interconnection (OSI) and non-OSI standards with a focus on the upper layer protocols. The protocols selected from the OSI stack are the Message Handling System (MHS), File Transfer, Access and Management (FTAM), and Transport Layer protocols. Those selected from the non-OSI standards are the Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), and Transmission Control Protocol/Internet Protocol (TCP/IP) protocols. The MHS framing overhead for a randomly chosen sample is developed in detail to provide an insight into one upper layer protocol header overhead in the OSI environment.

TABLE OF CONTENTS

1 Introduction....1
- 1.1 Protocol and Networking Trends....1
- 1.2 Scope....2
- 1.3 Analysis Methodology....4
- 1.4 Summary of Findings....6
- 1.5 Organization of the Report....8

2 MHS Framing Header Development....9
- 2.1 MHS Protocol Selection....9
- 2.2 MHS Protocol Overview....9
  - 2.2.1 Naming and Addressing....12
  - 2.2.2 Message Transfer Service....14
  - 2.2.3 Message Transfer Protocols....16
- 2.3 Typical Message Header....17
- 2.4 MHS Transfer Fields....25
  - 2.4.1 Routing Parameters. 25
  - 2.4.2 Conersion Parameters....27
  - 2.4.3 Delivery Time Parameters....28
  - 2.4.4 Security Parameters....28
  - 2.4.5 Content Parameters....28
  - 2.4.6 Miscellaneous Parameters....29
- 2.5 Message Header Fields....30
  - 2.5.1 Unused IPM Parameters....31

3 Selected OSI and non-OSI Protocol Characteristics....33
- 3.1 OSI Protocol Overview....33
- 3.2 Upper Layer OSI Protocols....34
  - 3.2.1 MHS Characteristics....34
  - 3.2.2 FTAM Characteristics....37
- 3.3 Lower Layer non-OSI Protocols....39
- 3.4 Upper Layer non-OSI Protocols....39
  - 3.4.1 SMTP Characteristics....40
  - 3.4.2 FTP Characteristics....41
- 3.5 Lower Layer Non-OSI Protocols....41

4 Frame Header Overhead Estimates....42
- 4.1 MHS Overhead 42
- 4.2 FTAM Overhead....43
- 4.3 Transport Layer Class 4 Overhead 44
- 4.4 TCP Overhead....44
- 4.5 SMTP Overhead....45

4.6 FTP Overhead 46
4.7 Lower Layer Overhead 47

5 Analysis Modeling of Frame Overhead 48
5.1 Analysis Modeling....48
5.2 Modeling Assumptions....48
5.3 Model Description 50
5.4 Analysis Findings....54

6 Conclusions....63
6.1 Frame Overhead Comparison....63
6.2 Protocol Overhead Optimizations....66

LIST OF FIGURES

1-1 Contemporary Network Configuration....7
2-1 Functional Configuration of MHS....10
2-2 Example of Transfer Header Fields....13
2-3 MTA Message Transfer Services and Protocols....15
2-4 Sample Message....18
2-5 Partial Transfer Envelope Encoding....20
2-6 Partial IPM Header Encoding....22
2-7 Message Transfer Fields and Their Subsets....26
3-1 GOSIP Version 2 OSI Architecture....35
3-2 Communication Entities in OSI Protocol Layers 36
3-3a Encapsulation of PDU Across All Layers....38
3-3b Generic Encapsulation of PDU....38
5-1 Sample OSI Internetworking Environment....49
5-2 Sample Analysis Model for Overhead Determination....51
5-3 Message vs. Overhead Increment....56
5-4 Header Variations with Message Size Variations....57
5-5 Message vs. Overhead Increment....58
5-6 Header Variations with Message Size Variations....59
5-7 Message vs. Overhead Increment....60
5-8 Header Variations with Message Size Variations....61
6-1 Representative Header Comparison for a 200 Octet User Message....64
6-2 Representative Header Comparison for a 20,000 Octet User Message....65

APPENDIX A....69
APPENDIX B.... 80

Vita....89

SECTION 1

INTRODUCTION

1.1 PROTOCOL AND NETWORKING TRENDS

The advent of improved wide area networking hardware along with the highly reliable, high-speed transmission media has brought about a proliferation of private data as well as voice and video networks especially in organizations that are spread geographically and where the need for reliable but inexpensive mode of inter-site communications is significant. Also included in this environment is the pervasive growth of local area networks. These trends have resulted in the preponderence of network-specific protocols.

The development of these multitude of protocols has been until now driven more by the applications and the clout of certain product groups than by the need for interoperability among networks. It was not until ARPANET came into existence that a *global* network connecting hosts and terminals on local networks was established and voluntary standards and procedures to regulate the allocation of addresses were created. These standards were later identified as Internet protocols - more commonly known as the Transmission Control Protocol/Internet Protocol (TCP/IP).

Over time, other groups joined in the development and further enhancement of these protocols which are now described in a set of documents called the Request for Comments (RFCs). On the international scene, the International Consultative Committee for Telephone and Telegraph (CCITT) spearheaded the development of communication standards needs for public networks under the auspices of International Telecommunication Union (ITU). The OSI standards are thus often adopted from the CCITT standards. Today, there are some two hundred technical committees working

and coordinating these standards within the International Organization for Standardization (ISO).

The TCP/IP protocols are not universal in that they are not used by the Postal Telegraphique Telephonique (PTT) type of public networks in most countries. Indeed they serve the Department of Defense (DoD) and certain other organizations that have volunteered to become member users. The ISO standards are, on the other hand, extraordinarily complex, uneven and incomplete. They are uneven because the layers do not have either the same size or significance. Some layers are ill-defined and are almost empty while others incorporate too many existing protocols and maintain a measure of redundant networking parameters. Additionally, the original OSI standard completely ignored the connectionless services and connectionless protocols - a criteria almost all Local Area Networks (LANs) use today.

Inspite of all these interoperability problems and shortcomings of standards, the desire for interoperability has only been growing stronger. The hardware and software vendors of communications and computers are forming groups and coordinating their intellectual efforts to provide more and more interoperability to the public. It is therefore only timely to investigate some important aspects of protocols that affect the interoperability issues related to these standards and explore the protocol optimizing possibilites that can be implemented without substantially changing the hardware configurations for the users.

1.2 SCOPE

This paper analyzes the functional and performance characteristics of the OSI and non-OSI protocols and related interoperability issues. The upper six layers of the OSI protocols are examined with emphasis on layer six and seven because the protocol

framing overhead in these layers can be significantly more than that in the other layers. The OSI protocols from layers six and seven reviewed are Message Handling System (MHS), and File Transfer, Access and Management (FTAM). The non-OSI protocols reviewed are (TCP/IP), Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP).

The performance issues are directly related to protocol characteristics especially frame header overheads. Header overhead includes additional data required for message addressing, control and accuracy. It is generally understood that greater header overhead will result in greater processing effort and subsequent reduction in throughput of a communication system. This study focuses upon the direct telecommunication implications for throughput due to the header overhead required by the protocols in use.

The selection of protocols, particularly in the Application layer, impacts most the amount of processing performed by the host processors, e.g., file servers, mainframes, et cetera. A significant amount of analysis was done in order to determine the overhead characterisitcs of one particular layer seven OSI protocol: the MHS. The MHS protocol is one of the most popular OSI protocols and more technical information is available on this protocol than other higher layer OSI protocols. Other applicable protocols were also reviewed within the scope of header overhead analysis. The results of all of these analyses are enumerated both quantitatively and qualitatively.

The research material was derived from the standards, technical literature and implementation tests performed in the recent past at National Institute of Standards and Technology (NIST) and other organizations. The representative selection of protocol framing architectures presented toward reviewing the contemporary networking environment and related protocol issues are generally the most widely discussed protocols. It must be noted that this selection and the analytical approach are

constrained by limited scope of this study; the selection does not represent any particular bias other than one of popular use.

1.3 ANALYSIS METHODOLOGY

An investigation of the current literature on protocols indicate that the overall performance considerations for networks and their underlying protocols can be segmented into first, the delivery system or the network system excluding the end systems, and second, the application processing system incorporating more commonly the end systems. This is true for networks that use non-OSI protocol standards as well as those that use OSI protocols.

Generally, the network performance criteria include such variables as the frame size of packets going in and out of each network segment, the seek time of each LAN server's disk, the minimum allowable delay for each LAN segement, the delay introduced by a router or gateway and the delay experienced in the Wide Area Network (WAN) between two gateways. For the internetworked environments, it is obvious that the slowest subnetwork component will determine the overall performance since it is here where the congestion or traffic bottleneck will first occur. Thus utilization is derived from the transmission rate of the slowest networking facility in the system.

One approach for performance evaluation would be to test utilization in a simulated (or an actual) environment replicating (or using) a preferred combination of network devices and applications. The network traffic characteristics could then be measured at critical points of the network - by and large at the bridges, gateways and the end systems and the performance could be derived for the several different protocols. But such research effort is beyond the scope of this study. So an alternative analytical methodology is chosen.

For the purposes of this theoretical study, a network configuration is first established and only the first parameter of performance evaluation above, namely, the framing overhead is selected. Since no standard exists for specifying and testing of network interface devices such as bridges, routers, gateways, or file servers, the performance comparision based on framing overheads is one reasonable way of assessing end-to-end performance through selected, large multi-segment networks of known applications.

In the simplest configuration, a contemporary network would consist of a number of LANs connected possibly to a backbone Fiber Distributed Data Interface (FDDI) ring. Two such FDDIs would communicate with each other over a WAN via a gateway at each end. Although configuration of this network environment is considered to be typical, the configuration of the LAN may vary from implementation to implementation and that of the gateway, from vendor to vendor. For the purposes of this study, a workstaion or fileserver, residing on a LAN, communicates with a remote workstation or fileserver via the WAN. The remote fileserver's connection to the WAN mirrors the local fileserver environment as shown in Figure 1-1

Processing of a message frame is a function of the *total* message content which includes the header information. While user data is implementation independent, the header information at various layers will vary with the network interface devices and protocols. Thus bridges and routers operate at the lower levels and the gateways or more commonly the end systems handle the applications. The performance is affected by the addressing schemes, number of message recipients and the kind of options selected. Additionally, the performance also depends on the type of links and switching facilities, the type and number of messages (or files or transactions) transmitted including the type and number of sources and destinations and how they are addressed.

In this study, one particular message is randomly selected with a typical addressing content. The framing overhead is developed for this message. The Message Handling Systems (MHS) protocol details are used to develop the message addressing header overhead. The header overhead for other protocol layers is also reviewed from information collected from results of analysis and experimentation carried out elsewhere.

Alternative framing options are reviewed in light of the current technological advances which provide very low-error, wide band communication media and fast processors. The detailed MHS overhead development is investigated to determine what procedures could be implemented to effect an optimized framing architecture.

The non-OSI upper layer protocols are also reviewed. The results of all overhead investigations are summarized in tables and graphs to show a comparative mesure of the overheads.

1.4 SUMMARY OF FINDINGS

The basic findings of this study relate to both the OSI and non-OSI protocols. It is determined that the frame header overhead imports for the upper layers in the OSI environment would be significantly higher than in the non-OSI environment. This finding is in line with the common expectations. The OSI standards attempt to serve a much wider range of users and provide a larger number of functionalities than the more *service-restrictive* non-OSI protocols such as the Internet protocols.

It is also determined that the protocol frame architectures can be used in an optimum fashion in order to attain a significantly low overhead without violating the standards while maintaining most of their benefits. The detailed analysis of MHS presented in this study and related investigations also demonstrate the following understandings.

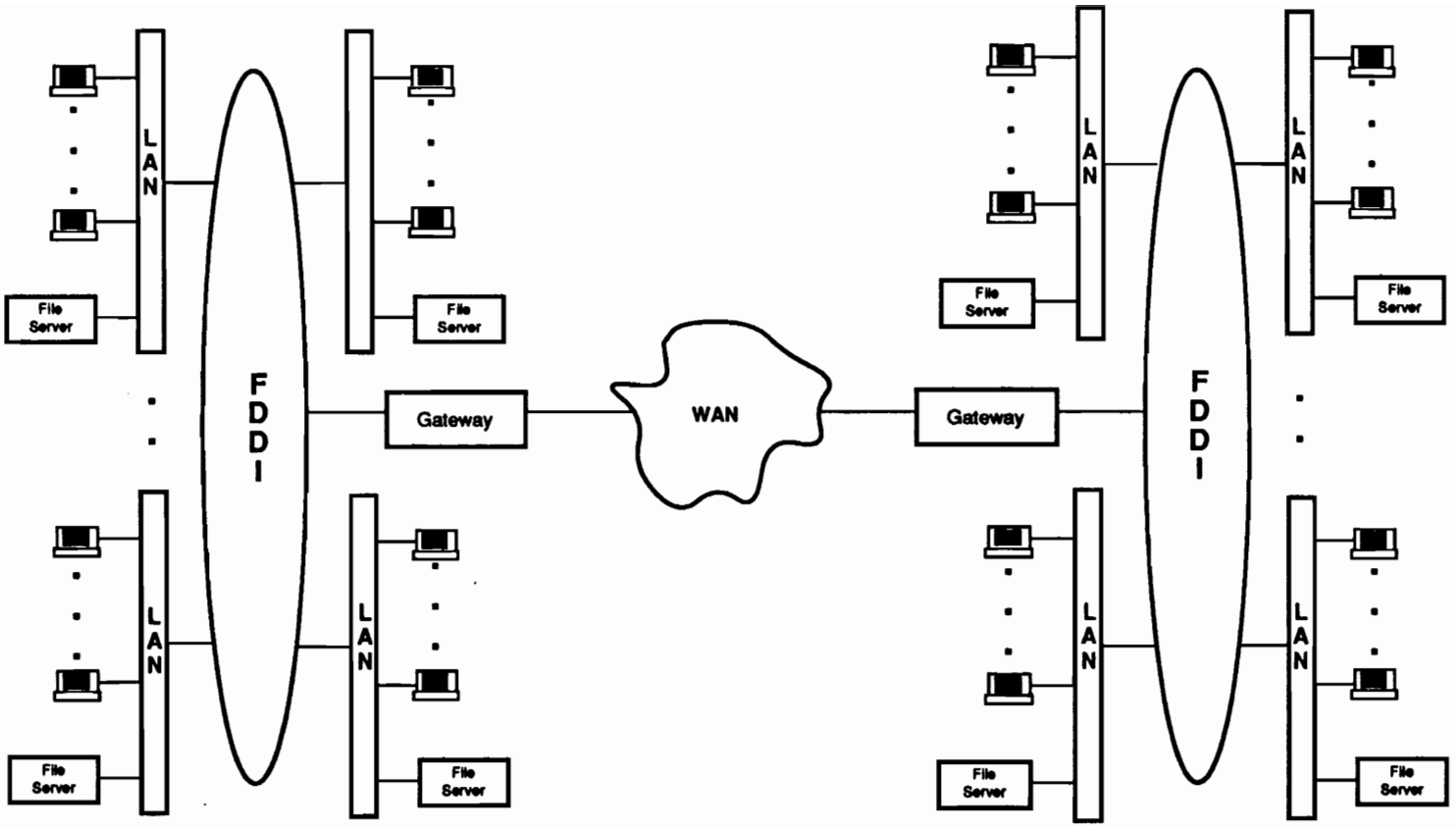


Figure 1-1
Contemporary Network Configuration

The upper layer protocols such as MHS and FTAM have the flexibility to allow tailored implementation in *limited usage* approach which can reduce overhead and processing delays. In this approach, the optional parameters and the addressing schemes are to be carefully chosen and implemented based on the network and message routing needs.

1.5 ORGANIZATION OF THE REPORT

Section 1 describes the current networking environment and provides an introduction to the scope. It also provides a short discussion of the methodology adopted for protocol analysis and a brief summary of the major findings and conclusions. In Section 2, the MHS header overhead of a sample message is developed in detail which is followed by a brief description and performance review of a collection of the OSI and non-OSI protocols in Section 3. Section 4 provides a comparison of performance and other characteristics of the protocols discussed in Sections 2 and 3. Section 5 describes the analysis modeling tool for layer by layer overhead variations for difeerent message sizes. The approach that one could take to optimize the higher level protocol architecture concept for maximizing performance and ease of implementation is also discussed in the concluding Section 6.

Appendix A provides the details of the Message Transfer Envelope (MTE) encoding and Appendix B provides similar encoding details for the Internal Personal Message (IPM) header.

SECTION 2

MHS FRAMING HEADER DEVELOPMENT

2.1 MHS PROTOCOL SELECTION

For the reasons stated earlier, the MHS is selected as the upper layer protocol. Its performance is measured in terms of header overhead it adds to the basic data message. The protocol data elements and the encoding procedure is developed in detail in this chapter. The overhead import to a typical message sample randomly chosen is determined. This determination also brings forth the overhead contribution by the presentation layer and provides insight into the protocol service elements and performance options available.

2.2 MHS PROTOCOL OVERVIEW

The MHS protocol as described in CCITT X.400 - X.420 recommendations actually comprises a series of protocols that specify how messages are transmitted and received from an originating user system to a receiving user system. In the language of the MHS protocol, a user can be either a person or a computer process.

The routing of the electronic mail takes place via several message switching or routing devices which are known as the Message Transfer Agents (MTA) and collectively form what is known as the Message Transfer System (MTS). The interface computer program between the user and the network is called User Agent (UA). A Message Store (MS) stores messages until the user is ready to receive them. The collection of the interconnected UAs, MSs, and MTAs is called the MHS. See Figure 2-1. The Access Unit (AU) facilitates access to communications services.

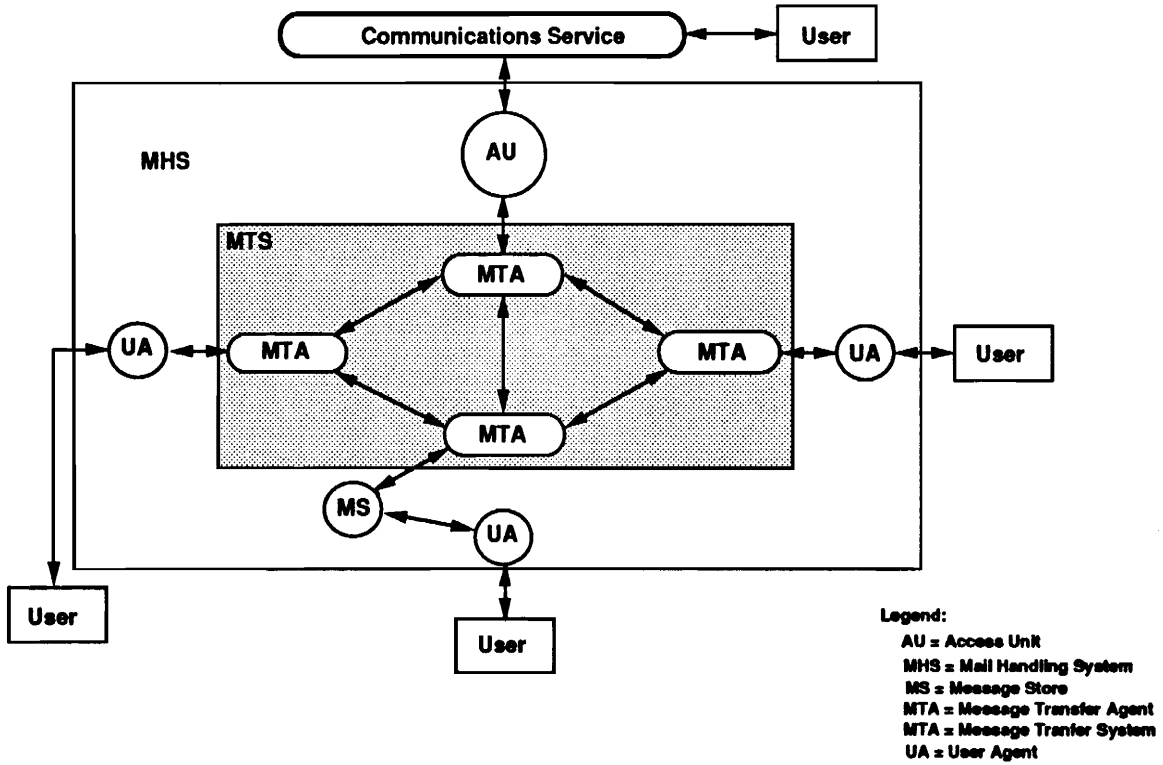


Figure 2-1
Functional Configuration of MHS

The conversion and reformatting of user's data is commonplace in almost all non-compatible communicating systems. The OSI approach uses a standardized and limited set of syntax conventions to describe data structures and data transfer operations between different machines and user applications. The Abstract Syntax Notation 1 (ASN.1) is used to describe the data structures. The transfer syntax is used for formatting the bit stream which is an unambiguous representation of the data with a predictable frame overhead for a message. The ISO 8825 as the specification for bit-level description or representation of data is called the Basic Encoding Rules (BER) and is used for encoding of data. In CCITT Blue Books, X.208 specifies the ASN.1 language and is aligned with ISO 8824 and its Addendum I; the X.209 describes the basic encoding rules and is aligned with ISO 8825 and its Addendum I.

In order for a communication device to be able to process data received from the user, it must first know the type of data. Each piece of information (data) has a *type* or datatype and *value*. The former defines the class of information, such as integer, Boolean, octet string, bit string, et cetera, in the *primitive* category, and Sequence (ordered set), set (unordered set), choice, et cetera, in the *constructor* category. The value is an instance of the type, such as a number or text. In the communication device, typically, an encoding routine outputs an optimized, self-identifying bit or octet stream based on the information on data type and the Application Protocol Data Unit (APDU) itself.

A single character (one octet) *tag* is used with each type to distinguish the different types in ASN.1. Each tag has a *class identifier* (ID) and a *number* such as UNIVERSAL 11 (which implies encrypted). The tag also identifies whether the data type or field is primitive or constructor. The tags regroup the types into four classes as follows.

(a) *Universal* - these are application-independent types,

(b) *Application-wide* - these are application-specific and used in other standards such as X.400, FTAM, etc.,

(c) *Context-specific* - these are specific to an application subset, and

(d) *Private-use* - which are reserved for private use (by other agencies)

The Universal class tag assignments are, for example, 1 for BOOLEAN, 2 for INTEGER, ... ,16 for SEQUENCE, 17 for SET, and so on, upto class type 29 for Reserved for Additions.

If no class name is specified with a tag, such as PRIVATE, it defaults to context-specific. All type names and module names begin with an upper case letter and all other identifiers start with a lowercase letter. Comments in ASN.1 are preceded and followed, the latter if required, by double hyphens (- -). A module name is an ASN.1 identifier (a sequence of characters) that identify the module.

The BER representation allows the data elements to be identified with one or more Type, Length and Value (TLV) combinations. An example is provided below. The BER applies to a large number of data elements and therefore BER representation of data is less efficient than EBCDIC or ASCII. See Figure 2-2 for an example of the transfer envelope fields.

Type	Length	Value	Type	Length	Value	Type	Length	Value
		Contents			Contents			Contents

2.2.1. Naming and Addressing

Message originators and recipients are defined in every messge by *OR name*, where O stands for the originator and R for the recipient. The OR name consists of two

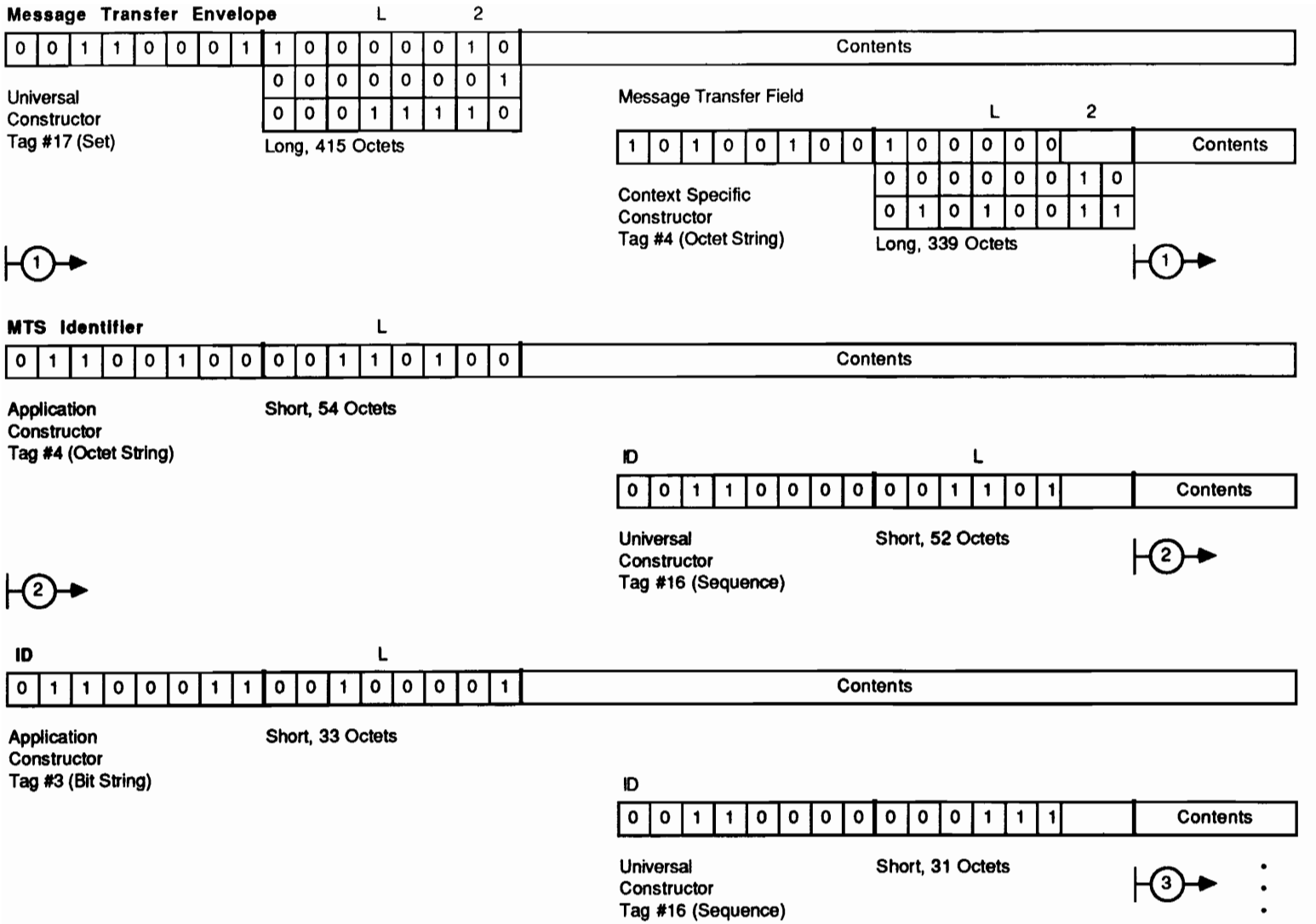


Figure 2-2
Example of Transfer Envelope Fields

components: a directory name and OR address. Atleast one of the two must be applied. The OR attributes used in this report are given below.

FIELD	CONTENTS
country-name	"301" (for U.S.)
administration-domain-name	"ADMD-Name"
private-domain-name	"PDMD-Name"
organization-name	originator within ADMD or PRMD
oganizational-unit-names	(as shown in the sample)
personal-name	not used
common-name	

2.2.2. Message Transfer Service

Although a review of all protocol details of the MHS access procedure is not within the scope of this report, some of the more essential service elements that allow access to MHS components are mentioned here for reference. In the OSI vernacular, a *Real Open System* consists of a component called Application Process (AP) which is a representation of elements that perform the application process. Such an element is the Application Entity (AE) which provides a set of communication capabilities. The AE in turn consists of the User Elements (UEs) and the Application Service Elements (ASEs) as shown in Figure 2-3.

The Message Transfer Service Element (MTSE) undertakes the role of transferring messages with the help of two service elements called the Reliable Transfer Service Element (RTSE), and Association Control Service Element (ACSE).

Message Administration Service element (MASE) supports administrative functions

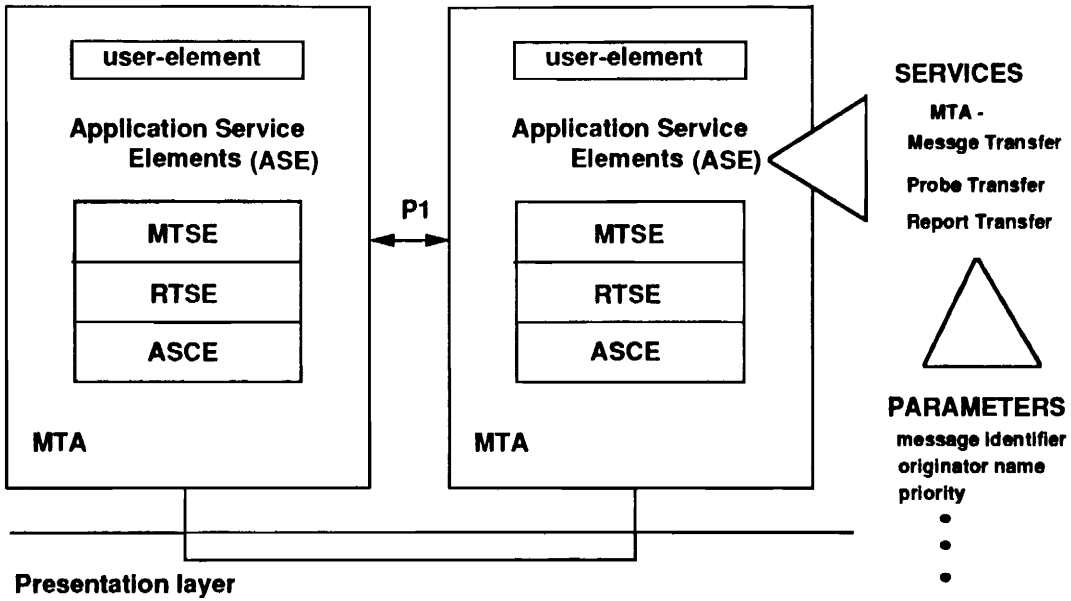


Figure 2-3
MTA Message Transfer Services and Protocols

among the UAs, MSs, and MTAs, and controls subsequent interactions with the help of the ASEs above. These service elements do not affect our framing definitions. These elements, acting as users, help the protocol framing conceptualization.

2.2.3 Message Transfer Protocols

The primary building block of MHS is called the Management Domain (MD). It contains atleast one Message Transfer Agent (MTA) and more than zero UAs. When such a domain is controlled by a public administration, for instance a PTT type of organization, the MD is identified as an Administration Management Domain (ADMD).

In case of a privately managed organization it is called a Private Management Domain (PRMD). A global concept is followed in this study which consists of both domains: a PRMD within an ADMD.

In MHS, the MTA checks every message for syntax problems after receiving it from the UA or AU. If the message is found free of problems, it is either sent to the local UA, AU or another MTA. There are three types of information conveyed by MHS. These are: *message*, *probe* and *report*. A message, only between users, consists of the *envelope* with the address - as in case of postal mail. The information *inside* the envelope is called the *content* - also like the regular mail.

The information on the syntax and semantics of the message is in the envelope and it is identified by the ASN.1 Object Identifier or Integer. Probe only contains the envelope - it is conveyed between the user (ultimate source) and MTA (ultimate destination). In both cases, the deliverability is checked by the MTA. The report, also between the user (ultimate destination) and MTA (ultimate source), indicates the status of delivery of the message or probe. In this report, only the message with appropriate options are dealt with for simplicity.

A set of primary X.400 protocols applicable to the analysis here are summarized below.

- P1 - is the routing or switching protocol applicable between the MTAs consisting of MTA bind and unbind operations and transfer port operations
- P2 - specifies the interpersonal messaging protocol, Inter-Personal Message (IPM), provides the heading fields for a specific MHS message
- P3 - specifies the MTS access protocol applicable between the MTA and a remote UA or MS
- P7 - specifies the MS access protocol used between the MS and a remote UA or MS

2.3 TYPICAL MESSAGE HEADER

A representative encoding of a randomly selected message is presented here. The message is displayed in Figure 2-4. No statistics were available for this study to determine the most common type of X.400 message traffic on MHS. This message was chosen with some inspection of available material on-hand and then tailored to be more representative of common office messages.

An important feature of the message is that it is addressed to four recipients in four different locations. It is a textual message with alphanumeric notations. The message has a qualifier not-urgent - which is the norm in case of MHS messages. This feature is not shown on the messsage but it adds minimally to the overhead import as will be shown later. The BER transfer syntax is used to code the message and thus generate the transfer envelope used by the transfer protocol and also the IPM heading, including most of the options. The assumption inherent in this approach is that the UA is

(Message Sample Description)

FROM: VICE PRESIDENT, ENGINEETING, TCPIP CO., WASH DC

TO: VICE PRESIDENT, ENGINEERING, FTAM CO., NEW YORK, NY

CC: VICE PRESIDENT, SALES, HQ, TCPIP CO, WASH DC

VICE PRESIDENT, ACCOUNTING, TCPIP CO, WASH DC

VICE PRESIDENT, ADVERTIZING, TCPIP CO, WASH DC

DIRECTOR, ENGINEERING, TCPIP CO, WASH DC

SUBJECT: SALE OF INTELLIGENT FUSION SPLICERS

1. AS YOU WILL RECALL, FTAM SUPPLIED AN ASSORTMENT OF 2000 (TWO THOUSAND) SUBJECT SPLICERS ON 16 MAY 1992. ALL BUT 180 (ONE HUNDRED EIGHTY) OF THE SPLICERS HAVE FAILED THE CQ-4 TESTS.

2. AS AGREED AT THE EARLIER MEETING, WE ARE HOLDING A FOLLOW-ON MEETING IN ROOM 803, BLDG 165, WASHINGTON, DC, AT 1330 HOURS, 04 JUNE 1993. EACH ACTION ADDRESSEE SHOULD BE REPRESENTED AND PREPARED TO DISCUSS THE ISSUES RELATED TO THE SPLICER PROCUREMENT.

3. WE LOOK FORWARD TO SEEING YOUR SPLICER PRODUCT DEVELOPMENT TEAM IN THIS MEETING AND TO RESOLVING ALL OUTSTANDING ISSUES.

(The message continues on...)

Figure 2-4

Sample Message

colocated with the MTA and the primitives associated with P3 and P7 can be eliminated. For instance, primitives such as *SUBMIT.Request*, *SUBMIT.Confirmation* and *DELIVER.Indication* are not required. The overhead import by using the UAE and MTAE connectivity protocol would not be high enough to effect the findings of the analysis in any significant way.

A typical representation of the transfer envelope and IPM fields is given in Figures 2-5 and 2-6. The actual ASN.1 representation and the related encoding are of minimal significance in this study as long as the overheald fields are accurately assessed. Thus each tag, irrespective of the Class and the data type, always adds an octet to a data element in the primitive data constrct, and to each content item line in the constructed data construct. Detailed TLV encoding of the message transfer headers and IPM headers are however provided in Appendices A and B for bit counts.

In Figure 2-5, transfer field encoding format is shown. The octets or bytes are read from left to right and top to bottom. The field identification is generally provided above each field. The first leftmost octet identifies the tag. The leftmost two bits identify the tag type in pairs of bits as shown below.

00 UNIVERSAL

01 APPLICATIONS

10 CONTEXT-SPECIFIC

11 PRIVATE

The next leftmost bit in the transfer field defines constructor type with a value of 1, and primitive, with a value of 0 (zero). The remaining five bits specify the tag number, for example, 11 for Universal 11. The next octet identifies the length of the data content

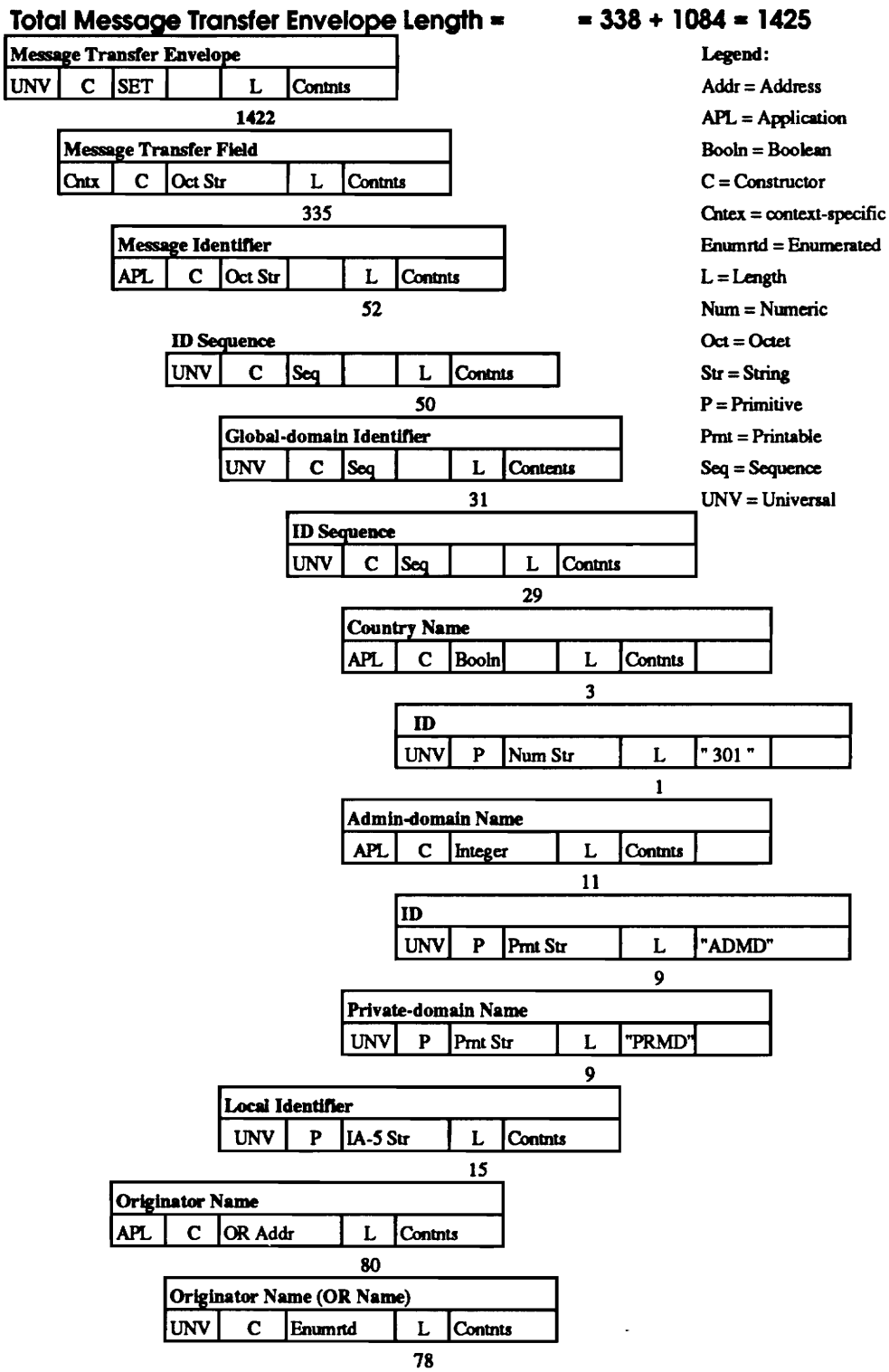


Figure 2-5
Partial Transfer Envelope

(Transfer Envelope continued)

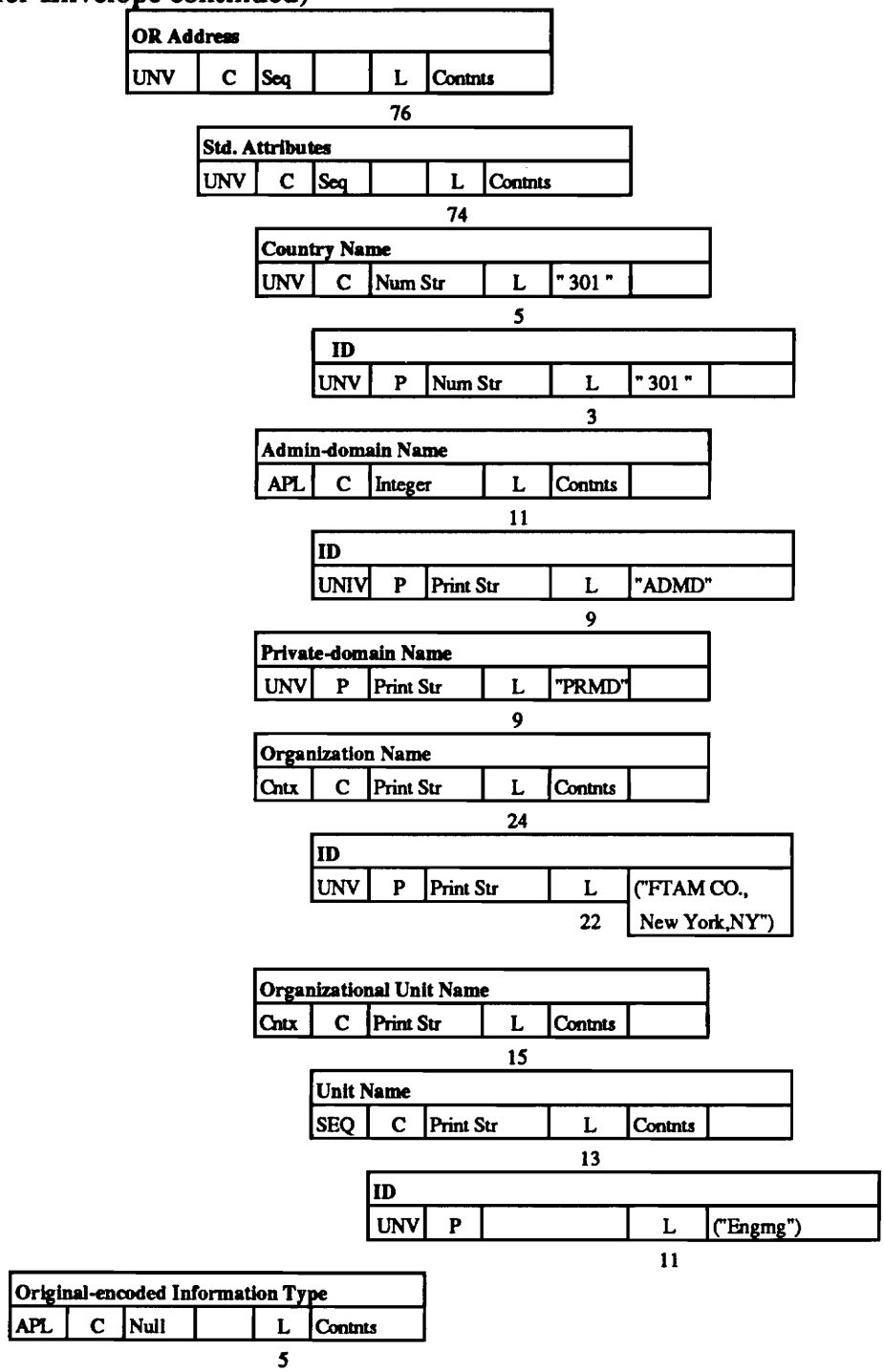


Figure 2-5 (concluded)
Partial Transfer Envelope

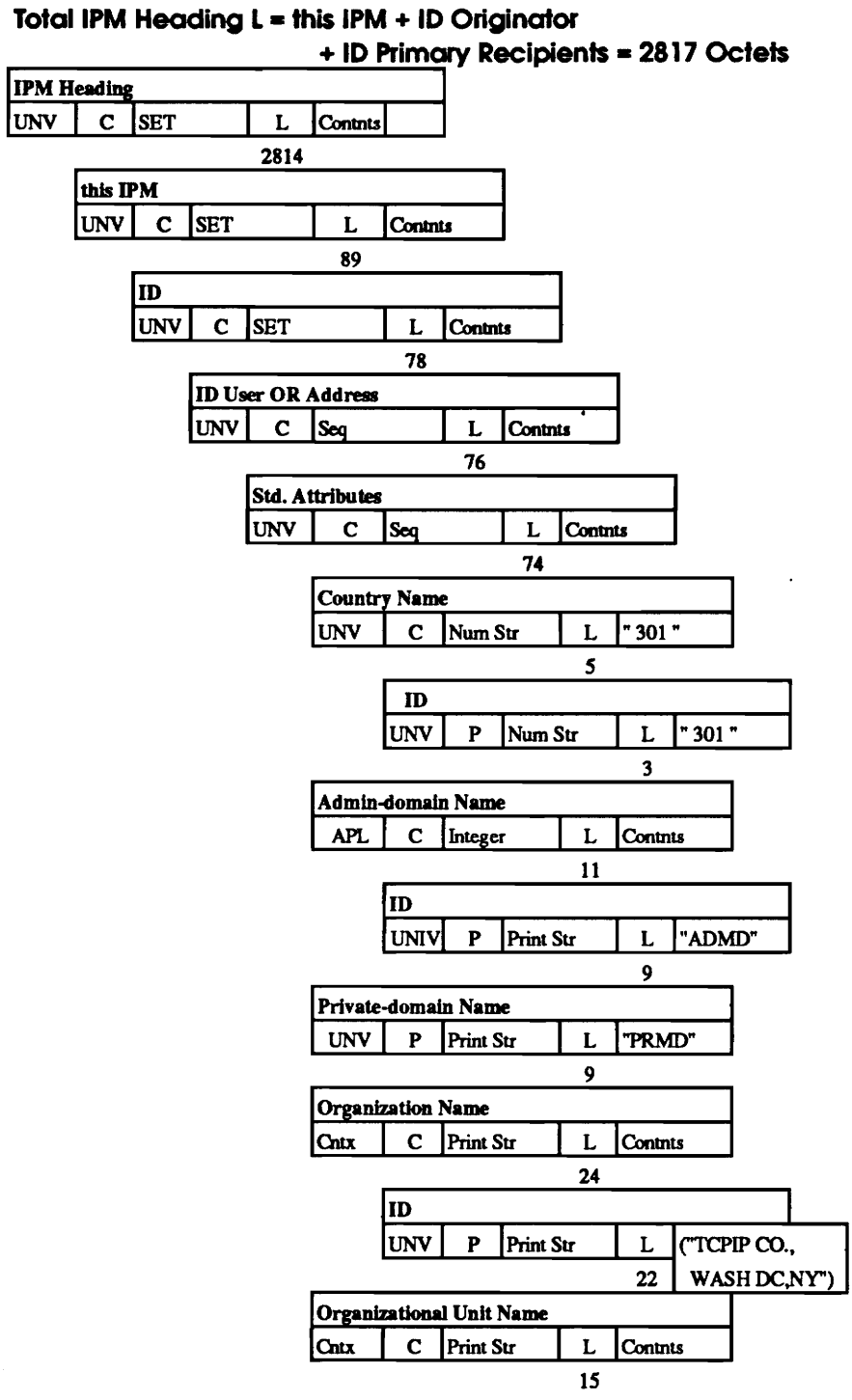


Figure 2-6
Partial IPM Header Encoding

(IPM Header Continued)

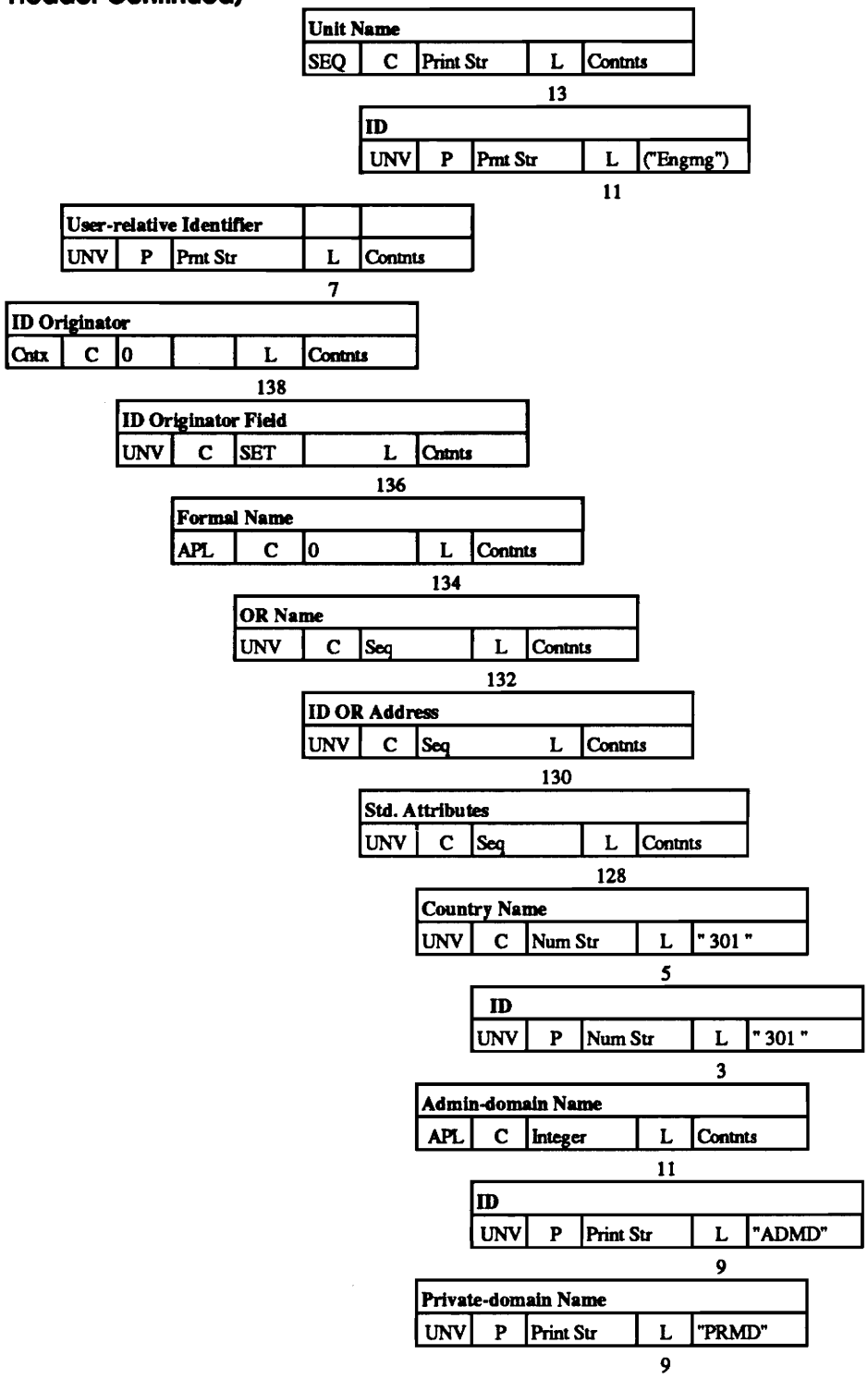


Figure 2-6 (concluded)
Partial IPM Header Encoding

of the field in number of octets. If the content field has a length shorter than 128 octets, only one octet is required to specify the length. But if the content field is equal to or longer than 128 octets, multiple octets are used.

In case of the multiple octets, the first octet has the high order bit set to 1 and the remaining bits show how many length-octets follow. The remaining octets specifying the length has each of their high order bit set to 0 (zero). Thus, a length of 52 octets (short one octet length) is simply 00110100, whereas a length of 339 is specified with a string of bits as 10000010 00000010 01010011 in all as shown in Figure 2-2. An indefinite form is also allowed by the standards. In this case, an end-of-contents (EOC) element is used to identify the end. The indefinite form has not been used in this report in order to optimize encoding.

If the word "CONTENTS" is shown following the length-octets, as in constructor type, it means the subsequent field octets identify the content of the data subfield as in the primitive type.

In the example shown in Figure 2-2, Universal MT envelope of length 339 octets is built with MTS identifier of 54 octets and several other fields. Both the MT envelope and the MTS identifier are constructor type; the latter is further built with a series of subfields.

The full catenated representation of the selected message's transfer envelope and IPM header are given in Appendices A and B. The subset of each field is staggered to the right while the fileds of the same parameter order are aligned on on the left. Encoding in bits for length and constructor data content has no real significance for frame header length determination, not does the tag type identification since it always is one octet long. The length is therefore shown as L with its decimal value just below it. The content in the constructor level field is shown as contents.

2.4 MESSAGE TRANSFER FIELDS

The message fields can be conceptualized as made up of two distinct types: the envelope type and the content type. The fields in the envelope type can be broken into two categories, first, the per-message-transfer fields, and second, the per-recipient field. The former field applies to P1 envelope as whole, while the latter to each recipient listed on the P1 envelope for a given message transfer. See Figure 2-7.

In this study, the encoding is broken into these two fields. The set of common parameters which are used in this report for the MHS transfer envelope are listed below. These parameters control the transfer, delivery, security and content of the user message and are crucial for the successful functioning of the MTA. Parameters defining redirection and conversion of messages have been avoided for simplicity. The parameters are grouped by several categories with assumptions for header overhead as determined through BER encoding.

2.4.1 Routing Parameters

These are also called the relay parameters. An applicable grouping of relevant routing or relay parameters is shown below.

a. Message-identifier: This consists of global domain identifier with country name, ADMD-name and PRMD-name and the local identifier which identifies the message - this is essentially a string of ASCII characters (actually IA-5 string). Values of 11 to 13 octets are assumed.

b. Originator-name: This field uses the OR address. It is assumed that personal name is not needed for the message and the average length of the field is 80 octets.

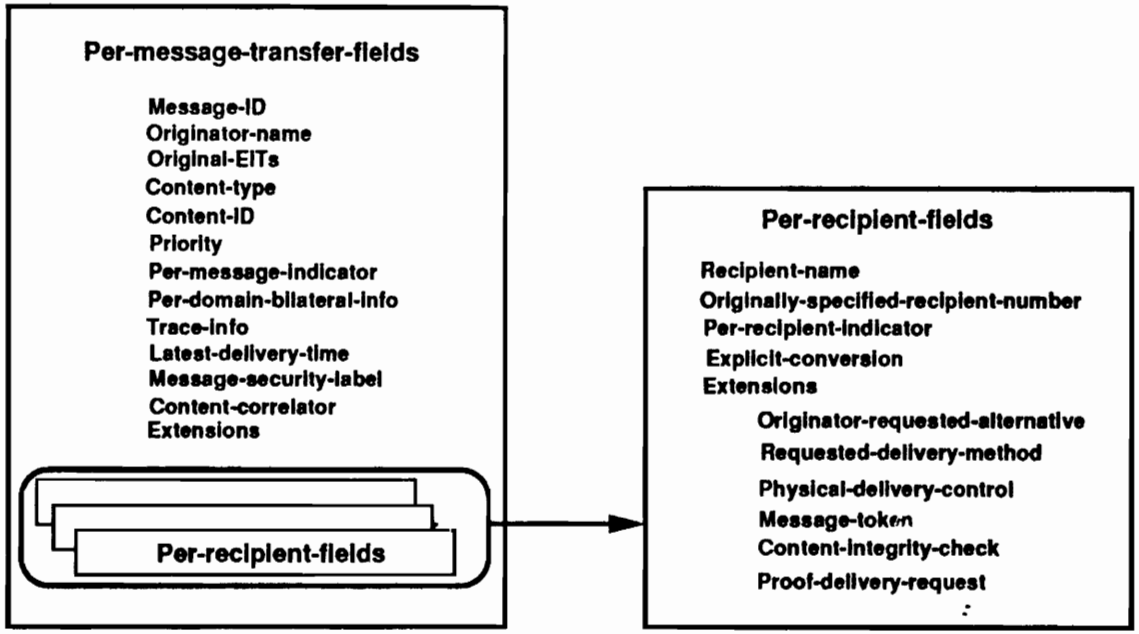


Figure 2-7
Message Transfer Fields and Their Subsets

c. Priority: MHS allows three options: non-urgent, normal and urgent, the overhead for non-urgent is used although mapping of other options will not affect the overhead. It is determined to be a 5 octet overhead.

d. Recipient-name: Since the standard allows only upto four organizational-unit-names, it was not possible to code more than four names of receiving organizations. This overhead is similar in size to the originator name overhead.

e. Trace-information: This is the overhead added by each MD and can therefore become substantial in the event the message traverses through several MTAs. For simplicity, it is assumed that the message traverses through only one MTA; overhead import is about 26 octets.

f. Original-specified-recipient-number: This specifies a distinct integer (starting from one) number for each recipient. This parameter along with the message-identifier unequivocally identifies the copy of the message to be delivered to each recipient. It is assumed that this number is good for one copy; a value of 5 octets is determined.

g. Extensions: This is used for adding options or more header fields. Generally, users tend to extend messages in these fields by indicating latest delivery time, security label and delivery method. A conservative value of 181 octets has been chosen. (No specific statistics could be found to determine what kind of extension header lengths are typical.)

2.4.2 Conversion Parameters

Two conversion parameters have been selected; these are given below.

a. Original-encoded-information-types (EITs): It is ssumed that the original message was encoded in ASCII (actually, in IA-5 string); the overhead is of the order of 7 octets.

b. Conversion-with-loss-prohibited: To facilitate an easier transfer of message, this conversion parameter has been allowed. This parameter prohibits conversion that entails loss of information. It's value is of the order 15 octets.

c. Explicit-conversion: This specifies if message conversion is needed (e.g., from facsmile to IA5-text). It is assumed this option is not required, if used, it would add 3 octets.

2.4.3 Delivery Time Parameters

The delivery time parameter most commonly chosen is the latest-delivery-time. The parameter specifies a date and time after which the message is not to be delivered. This is preferred over the other parameter of deferred-delivery time since the latest time than the earliest time for delivery is considered to be applicable to most messaging. It would add upto 25 octets to the header field.

2.4.4 Security Parameters

The security parameters chosen is the message-security-label that allows the sender to attach a security sensitivity label to the message. There are several other security parameters designed to counter security risks. The overhead import is 17 octets.

2.4.5 Content Parameters

The following content parameters are included for discussion here. The content-correlator parameter is considered not directly useful for the overhead evaluation.

a. Content-type: This specifies the syntax and semantics for each message. It is assumed the content follows the 1988 MHS format; the overhead is message specific,

however, 3 to 5 octets seem to suffice in most cases as well as for the message adopted in this study.

b. Content-identifier: This identifies the subject of the message. It is assumed an average length of 10 to 12 octets would suffice since the content would vary from message to message.

c. Content-correlator: This parameter is ignored since only one message is dealt with in this study and no subsequent messages are involved.

2.4.6 Miscellaneous Parameters

Some of other significant parameters reviewed for use in this study are annotated below.

a. Per-domain-bilateral-information: This field contains transfer information not defined in the standard but whose usage is standardized within a particular domain. Since it is not defined todate, this field has been ignored.

b. Per-message-indicator: This indicator allows options that the sender can select for the message irrespective of who receives the message. For the message selected, four options, namely, (a) allow disclosure of recipients, (b) allow an alternate recipient, (c) allow implicit conversion, and (d) do not request return of contents, are allowed resulting in the total overhead of 5 octets.

c. Per-recipient-indicators: Selection of options under this category do not affect the overhead from one recipient to another. This overhead is of the order of 5 octets and varies with each recipient.

2.5 MESSAGE HEADER FIELDS

The IPM earlier identified as P2, is the only content-type standardized. P2 can be divided into two parts: header and content body. The overhead fields associated with the IPM header are listed below. They form a part of the message content and do not affect the transferring system. There are several other IPM header fields available in the standards. For reasons of simplicity, only the listed types are encoded for the selected message.

a. This-IPM: This field identifies the IPM with an identifier component. It consists of two parts, first the OR address of the originator, and second, a user-relative identifier. A total of abour 95 octets are required for this field.

b. Originator: Since an informal representation of the originator address is allowed, it is assumed that the OR address and telephone number will suffice for the OR Descriptor. The authenticated originator address can be taken off the transfer envelope fields (by the MTS) as described before; the total field length is approximately 120 octets.

c. Primary recipient: This field is used to identify intended primary recipients. The OR address is used in this field; the free-form-name should be sufficient to indicate the primary recipirent(s). The authenticated OR address could be taken, as stated before, from the envelope. The total field has a length of 509 or so octets.

d. Copy-recipients: The OR addresses are used for this field; the total for each of the four recipients is approximately 116 octets. Here, it is assumed the message is copied to four other recipients.

e. Subject indication: The subject line of the message is used in this field. This amounted to about 55 octets.

f. Expiry time: This option provides the notification for the message expiration time as authorized by the user. A date/time for this field is assumed. This field imports about 13 octets.

g. Reply time: This field imports about 13 octets of overhead which could be saved if this field is not used.

2.5.1 Unused IPM Parameters

Some of the other significant IPM header and body fields reviewed for this report are listed below but these fields are not used in this study. There are several other IPM service elements that have not been shown below because they have limited significance for this study.

a. Auto forwarded indication: This field is used to indicate if the message is a result of auto-forwarding.

b. Blind copy recipient indication: This field was not used in the message encoding.

c Importance: This field, which identifies importance of a message (low, high or normal), it is assumed, is not needed for our message since the urgent status is indicated in the message transfer field under priority. Overhead of 3 octets is thus saved.

d. Forwarded-IPM-indication: This field indicates that the message contains a forwarded IP part. This field was not used.

e. IP message identification: This field identifies the IPM and is used by the IPM UAs. The OR address and identifying number (the user-relative-identifier) have been used in the sample encoding.

f. Obsoleted IPMs: This identifies the IPMs considered to be obsolete by the user. It is considered to be of little significance.

g. Related IPMs: This field was not used since it is assumed no other IPMs are related to this. If used, the user-related-identifier component of the IPM-identifier would be adequate to provide the necessary information.

h. Replied to IPM: This identifies the message to which the present message is a reply. This field was not used since the message content could take care of this aspect of the message and thus save the overhead.

i. Reply recipients: This option identifies the recipients that need reply to the message. It was considered that such items would be more effective as part of the message. It is therefore determined that this field need not be coded allowing a saving as much as 260 octets in the overhead.

j. Sensitivity: This field is assumed to be unnecessary since there is nothing secretive about the message. About 3 octets are saved.

SECTION 3

SELECTED OSI AND NON-OSI PROTOCOL CHARACTERISTICS

3.1 OSI PROTOCOL OVERVIEW

This study uses the U.S. Government Open System Interconnection Profile (GOSIP) version of OSI protocols where applicable. In 1991, Federal Information Processing Standard (FIPS) 146 adopted GOSIP as a network implementation standard to facilitate the application of advanced technology and interoperability between multiple vendor systems. The GOSIP Version 2 OSI architecture is presented in Figure 3-1.

In OSI and as well in GOSIP protocols, the transfer of information necessary at any given layer is the responsibility of the lower layers. Each of the layers contains entities that exchange data and provide horizontal communications with peer entities as shown in Figure 3-2. Given below are brief version of each layer definition from FIPS 146.

The Application layer (layer 7) allows for protocols and services required by particular user-designed application processes. The Presentation layer (layer 6) specifies or, optionally, negotiates the way information is represented for exchange by application entities. This layer is concerned only with the syntax and not the meaning of the transferred data.

The Session layer (layer 5) allows cooperating application entities to organize and synchronize, and to manage data exchange. The Transport layer (layer 4), provides reliable, transparent transfer of data between cooperating session entities. This layer entities optimize the available network services to provide the performance required by the session entities. Together, these layers are called the upper layers.

The Network layer (layer 3) provides the message routing and relaying between end systems on the same network or on interconnected networks. This layer also provides hop-by-hop network service enhancements, flow control, and load leveling.

The Data Link layer (layer 2) provides communication between two or more (multicast service) adjacent systems. It provides frame formatting, error checking, addressing, and other functions necessary to ensure accurate data transfer. The Physical layer (layer 1) provides a physical connection for transmission of data between data link entities. This layer performs electrical encoding and decoding of the data. Together, these three layers will be called the lower layers.

A representative scenario of data packet construction down and up the seven layers of the OSI protocol stack is shown in Figure 3.3 (a) with a generic construction in Figutre 3.3 (b).

3.2 UPPER LAYER OSI PROTOCOLS

In this study, the upper layer header overhead considerations relate to the layers four through seven; the layers are Application, Presentation, Session and Transport. All of these layers are resident in systems such as the hosts, file servers or the gateways. The major protocol craracteristics reviewed in this report are MHS, FTAM in the Application and Presentation layers.

3.2.1 MHS Characteristics

As stated earlier, the MHS is not designed for real-time applications. However, one of the MHS quality-of-service (QOS) parameters allows three different grades of service or delivery requirements: urgent, normal and non-urgent. In CCITT F.410, the guranteed delivery times for the Public Administration Management Domain (ADMD)

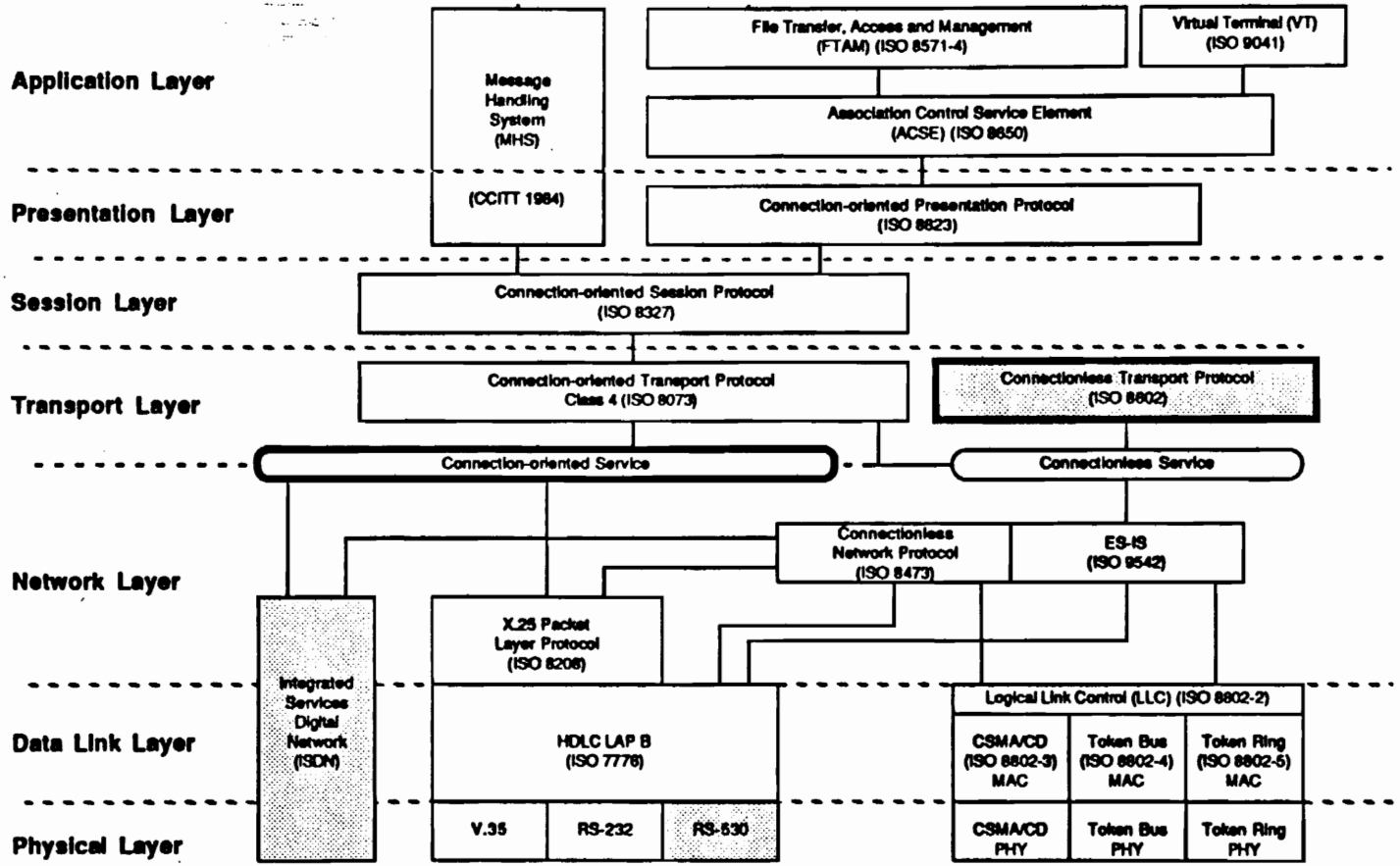


Figure 3-1
GOSIP Version 2 OSI Architecture

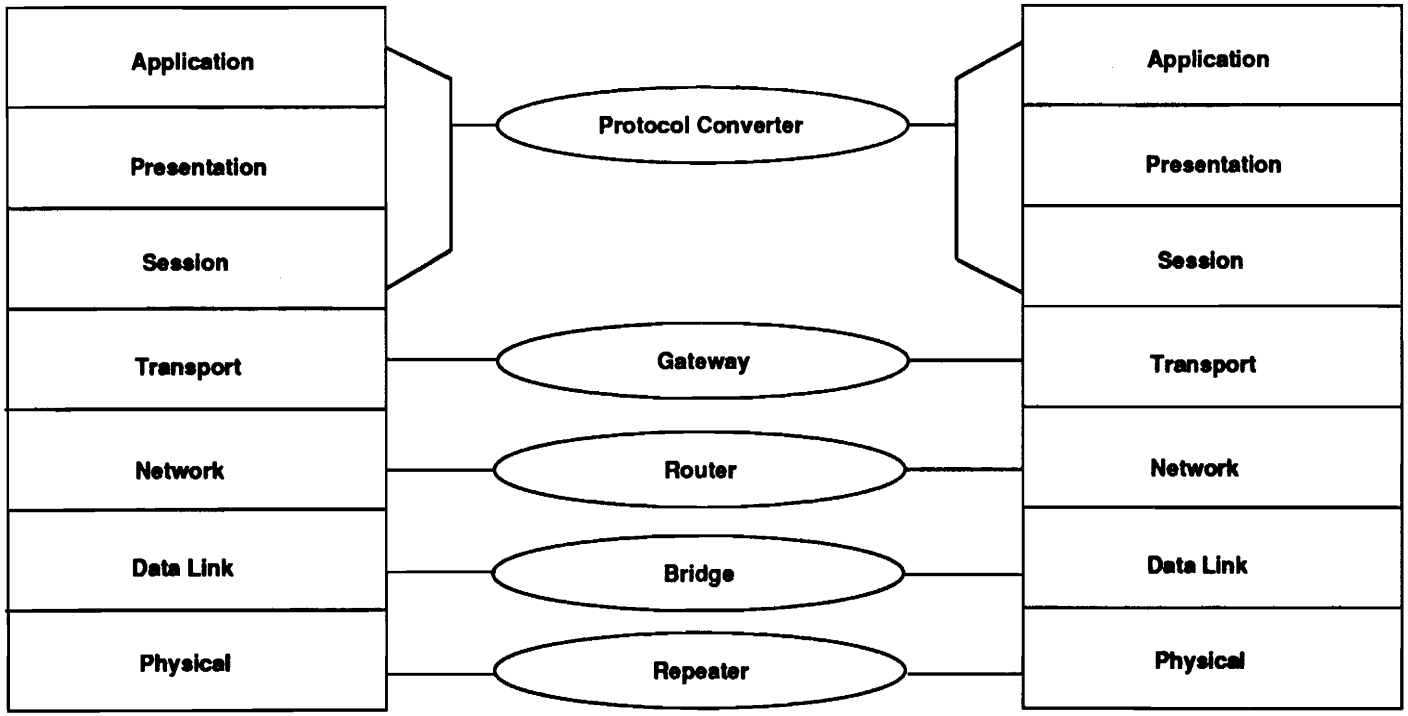


Figure 3-2
Communication Entities in OSI Protocol Layers

are specified. It is stated in this document that 95 percent of the messages are to be delivered within 45 minutes. For the delayed-urgent messages a non-delivery notification should be forced after four hours after submission. This delivery time can be improved to about 60 seconds - a much stricter delivery requirement - with grade of service modifications to the protocol

.

3.2.2 FTAM Characteristics

The FTAM protocol is designed to support the transfer, access and management of remote files. It works as an interactive system. It provides the capability to interact with a variety of internal file structures and can be operated in an interactive environment. Described in ISO 8571, the protocol has access control attributes that lets the user read from and write into the remote files. FTAM allows transferring files of varying internal structures and manipulating the records within a file.

FTAM is implemented in the mainframes, minis, file servers, and the gateway systems. All these systems or devices transfer, access or manipulate files on related remote systems. In case of general file transfers, delays or congestion control needs to be worked out and if required, timer driven capabilities need to be implemented to space the file transfer loads according to their delivery requirements. This is a network issue. Considering available time windows and large transmission bandwidths in today's networks, timer requirement is generally not necessary for file transfer.

FTAM requires the user to login, provide a password and identify location before access to the remote system is allowed. This is an option and is well-suited for the security-sensitive modern networks that span over many long-haul wideband links. The option could be automatically exercised by an application over FTAM. The FTAM phases, representing protocol exchange activities, are sequential. FTAM requires the use

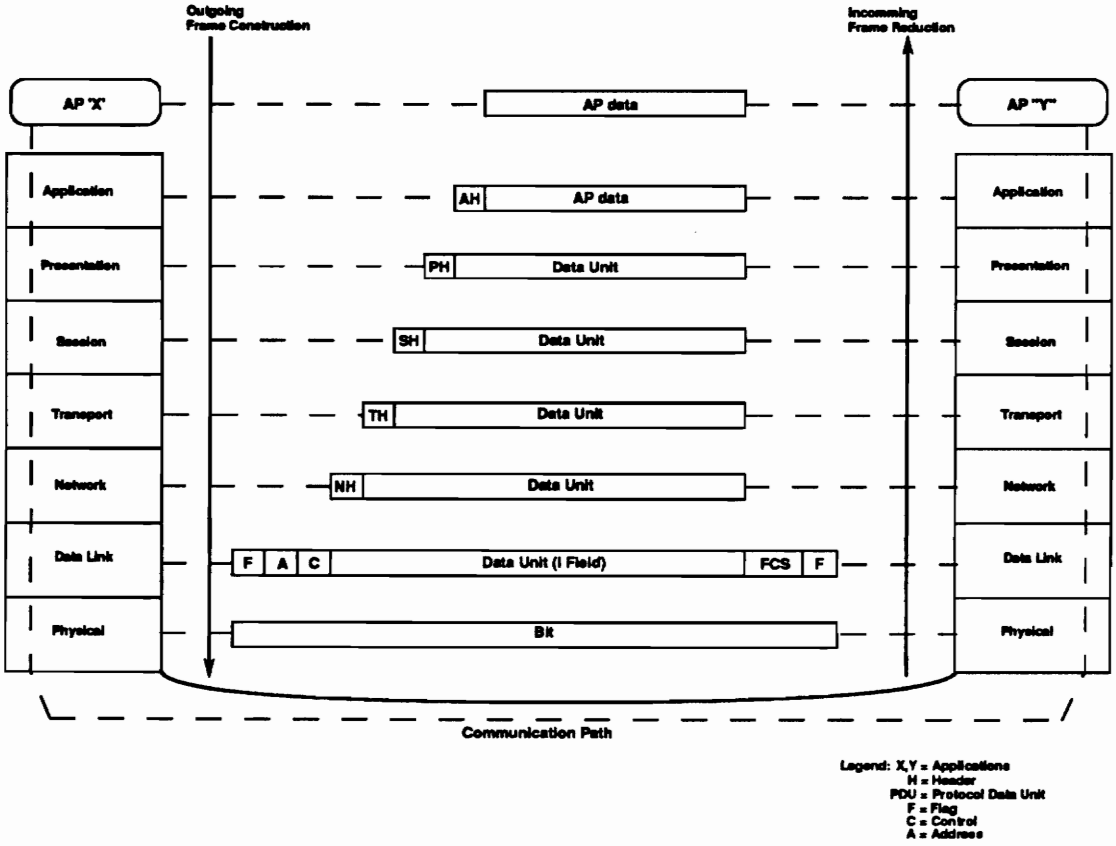


Figure 3-3a
Encapsulation of PDU Across All Layers

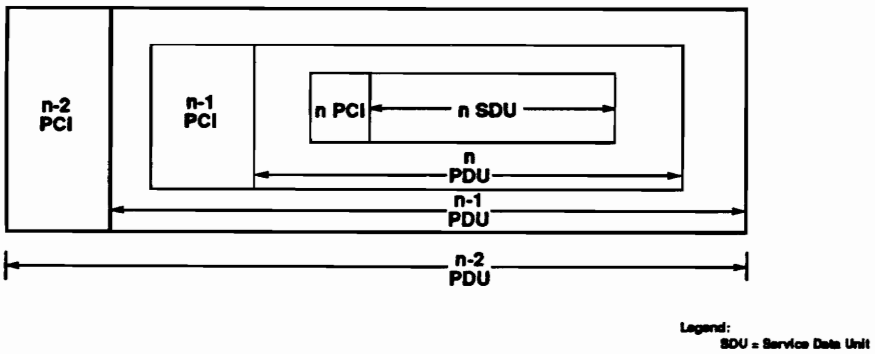


Figure 3-3b
Generic Encapsulation of Protocol Data Unit

of primitives and state diagrams during a phase. There are some 17 essential primitives, e.g., F-INITIALIZE, F-TERMINATE, F-DELETE, F-LOCATE, et cetera. Furthermore, each primitive has its own set of parameters.

3.3 LOWER LAYER OSI PROTOCOLS

The lower layer protocols involve network dependent variables. For the purposes of this study, layer 1 or the Physical layer is ignored altogether since the sample message selected for analysis is assumed to be unencrypted and since this layer would affect almost all messages equally for a given network environment.

3.4 UPPER LAYER NON-OSI PROTOCOLS

There is no presentation and session layers in the Internet protocol suite. However, file transfer and mail transfer tasks are done with the FTP and SMTP. The Telnet protocol provides for the remote login capability. Finally, the Internet Control Message Protocol (ICMP) is used for error and control messages for the datagram network.

The Internet protocol suite is a set of protocols which include TCP/IP . TCP/IP are the best known of the Internet protocols and therefore the whole suite is commonly called the TCP/IP family. IP is essentially a connectionless environment where information is sent or received as a sequence of "datagrams" (units of data), however, within this setup, TCP is a reliable, connection-oriented protocol. Each datagram is treated separately by the networks. The routing is done using an "Internet address" which is a 32-bit header representing four decimal numbers or four octets.

The TCP breaks up a message into datagrams at the near end and reassembles them in order at the distant or other end. The IP, on the other hand, takes care of routing the datagrams to the appropriate destination(s) and also fragments the datagrams where

necessary. Additionally, it maintains an updated log of all routing information. This, in most large networks, is a complex task. All these considerations have no impact on the header overhead.

TCP and IP are not affected by the character code representation (for example, ASCII versus EBCDIC) of the data. Many of the commands use standard ASCII which allows easy diagnostics. In general, Internet protocol commands use normal text. Sometimes a log file is maintained for easy auditing and human interaction. An IP datagram is different and generally larger than a packet. For instance, when TCP/IP is used on top of X.25, the X.25 interface breaks down each datagram into 128-byte packets.

3.4.1 SMTP Characteristics

The electronic mail pioneered by the ARPANET is widely used and is known as SMTP. This protocol is designed only for ASCII text. Other character sets or digitized information or picture is not allowed. There is no distinction between the envelope and the message (as in MHS). Each mail is a file containing header fields at the top. These fields are ASCII key words. Comments are shown in parentheses and ignored. Since the headers are a part of the message, they can be easily modified. However, the header fields are identified with a name and a three letter assigned domain name.

The header field increases as the message traverses each hop of the network with a short "receive" header information added for each hop. So, the tracking of the message routing is easy. The header field is a direct function of the number of network hops. The SMTP and MHS header formats are quite similar and mapping one to the other is relatively simple.

3.4.2 FTP characteristics

FTP is an Internet file protocol which primarily allows a user to transfer files from one computer to another computer. FTP is an application protocol which runs on top of TCP/IP. The message from the FTP layer is given to the TCP/IP which ensures its reliable delivery by taking care of the transmission and networking details. FTP is a utility that is used to access file on a remote system and then copy it to one's own system. This allows one to work with the file on the local system.

3.5 LOWER LAYER NON-OSI PROTOCOLS

There are no specific Internet lower layer protocols compatible with the lowermost two layers of the OSI protocol stack. The TCP is similar to OSI layer 4 Transport Protocol, Class 4, and IP is an OSI layer 3 equivalent protocol. The two lowermost layer protocols are therefore considered the same under both OSI and non-OSI standards in this report.

In the TCP/IP environment, TCP can "negotiate" the maximum size datagram that can be sent. The two ends decide on the lower of the two datagram sizes each can handle. However, none of them will know the restrictions imposed by the intermediate networks.

The IP takes each of the datagrams sent by TCP with the TCP header information and adds its own header. The main components of this header are the source and destination Internet addresses, the protocol number and another checksum. These are 32-bit addresses and look like *132.8.3.188*. The protocol number is an indication to the IP at the other end to send the datagram to TCP since there are other protocols that uses IP.

SECTION 4

FRAME OVERHEAD ESTIMATES

4.1 MHS OVERHEAD

When all the MHS IPM and the transfer fields were encoded, it was found that the total MHS overhead, inspite of all the selective optimization, is substantial. The encoding tables for the transfer envelope and IPM header were developed using Microsoft Excel software. The parameters were defined with interconnected values of related length octets and in a catenated fashion. The data values and types could therefore be easily manipulated to see increase or reductions in the header lengths. The following observations are made with regard to the encoding.

a. Over some 4200 octets were needed to encode the two sets of fields. Out of this, about 1400 octets were needed for the transfer envelope and another 2800 octets for the IPM header including header fields for the primary and for-info recipients.

b. If only one recipient is considered in the analysis, the total header overhead reduces to 1480 octets approximately.

c. Although the ORAddress is of the order of 76 octets in the example, it could be as high as 260 octets.

d. Each additional recipient added about 910 octets including the options. The total number of recipients was a major factor in the high value of the header overhead.

e. It was found that if in the model of transfer envelope, the originator and recipient "names" are reduced to 2 octets in lieu of 22 or so octets, and ADMD and PRMD lengths to one octet instead of 9 or so octets, as could be done in a numeric representation of address information, then the total transfer envelope reduces to about

600 octets and the IPM to about 900 octets. These reductions translate into 60% and 70% savings respectively in the full length header fields.

Further optimization, although to a lesser extent, can be realized through an elimination of unnecessary and redundant parameters. In Section 2, it was shown how some of the parameters were redundant and thus not used. It was also discussed how a reasonable optimization of data frame overhead in MHS can be attained with a selective addressing scheme and elimination of unnecessary parameters.

In the example under review, no Application protocol beside message transfer was considered for setting up connections and estalishing associations. The assumption made is that the Application Protocol Interface (API) contribution to frame overhead is negligible. Finally, the frame overhead did not inlude the actual data content of the message.

4.2 FTAM OVERHEAD

Current experimentations with FTAM on a limited scale have shown that in using FTAM protocol, the layer 7 header overhead is much higher at the initial set up phase. At the set up time, the header overhead for the top three layers may be of the order of 300 octets in which FTAM (layer 7) imports the majority part of about 160 octets while the presentation and session layers account for about 140 octets. The presentation layer itself accounts for about 100 octetss while the session layer, 40 octets.

As the connection is established and the FTAM activity progresses, the overhead demand settles down to about one third to one quarter of the above requirements. For the comparative effort in this study, we assume the worst case header overhead import of 300 octets.

4.3 TRANSPORT LAYER CLASS 4 OVERHEAD

Of the four classes of variations available for the Transport layer protocol, the header overhead contribution by the Class 4, which is designed for unreliable network service, is the most significant. The transport protocol data unit (TPDU) are of ten differnt types. Each TPDU consists of three header parts in addition to the user data.

The first part is 1 octet long and defines the combined length of the variable and fixed parts of the TPDU. The second part is the fixed part of the header and can vary from 3 to 7 octets depending on the kind of TPDU. The third part is the variable part of the TPDU and its length depends on the parameters associated with the message. It may contain any of some 14 options available including priority, option bits, and checksum.

The maximum length of the fixed and variable parts including the length indicator can be upto 254 octets. A typical layer 4 overhead breakdown shows that about 5 octets of data frame overhead is required in the connection-oriented mode. This excludes the large overhead of 27 or so octets for connection establishment, 5 for acknowledgement and 9 for data transfer with checksum.

4.4 TCP Overhead

The minimum TCP header size is 20 octets. The header consists of one header format, unlike the TP4 header, and includes a 32 bit sequence number, a 32 bit acknowledgement field, six one bit flags which are connection status related, a 16 bit checksum field, another 16 bit urgent pointer, and 0 to 32 bits for options for miscellaneous items such as the buffer sizes during the set-up orocedure.

A header of information is attached to each datagram by TCP. This header is at least 20 octets long with source and destination "port numbers" and a "sequence number"; the latter maintains the order of the datagrams upon receipt while the former keeps track of

the "conversations" between the machines. TCP does not actually number the datagrams but rather the octets of data in each datagram. The header also includes the checksum and acknowledgement numbers. The checksum allows TCP at the other end to ascertain that the header and data (TCP segment) is intact. On top of all this each TCP header has source and destination addresses encoded in 16 bits each.

4.5 SMTP OVERHEAD

SMTP treats each mail as a file and does not make any distinction between the envelope and the message. So, each piece of mail is a file with certain ASCII header fields. The RFC 822 specification was used to ascertain that the header fields are about 18. The same header overhead as in case of the FTP protocol given below is assumed to be applicable.

4.6 FTP OVERHEAD

The FTP commands specify the parameters for the data connection and the kind of file system operation required. FTP uses the Telnet protocol on the set-up or control connection. This is done in one of two ways. First, the user (client) and the server implement the Telenet protocol, or second, they make use of the existing Telnet module in the system. The first approach provides efficiency and independence, and since it does not involve a large amount of data, it is preferred to the second approach. However, it ought to be noted that FTP relies very little on Telnet.

The data connection is used for the file transfer only. Basically three commands are used for the transfer - Mode, Structure and Type. One of the three (data) structures called the page structure, for discontinuous files, has a minimum 4 octet header. The

other two, called the file structure and record structure, consider the file as sequential bytes and records respectively.

There are three modes of transmission. The first mode, known as the Stream mode, allows data to be transmited as a stream of bytes. Record structure is allowed with negligible overhead. The next mode is the Block mode in which the file is transmitted as a series of data blocks with approximately three header octetss. The third mode is the Compressed mode. which could entail a overhead of seven or so overhead octets. We assume a rounded up figure of 20 octets for the FTP overhead.

4.7 LOWER LAYER OVERHEAD

The lower three layers of the OSI protocols are the header overhead contributed by the Media Access Control (MAC), the Logical Link Control (LLC), and the X.25 LAP-B layers are assumed to be applicable to the OSI 8802.3 LAN environment under consideration. A total header overhead of 40 octets is assumed. This overhead may vary by a small margin depending on the network parameters.

In the LAN environment, due to large bandwidth availability, the header overhead considerations for FDDI are far less restrictive than the OSI 8802.3 LANs. Also, FDDI does not have the wide installation base as do the 8802.3 LANs. It is thus assumed unnecessary to examine the FDDI overhead import here.

In the non-OSI environment, the IP datagram header has a 20 octet fixed part. It also has a variable length part option field reserved for security and miscellaneous information. The maximum length allowed for both data and header is a little over 65K octets. One unique feature of this protocol is the one bit Don't Fragment (DF) field that can prevent fragmentation of the datagrams if the receiving end is not capable of

regrouping the datagram pieces together. In the header overhead analysis, an estimated value of 25 octets appears reasonable as the IP header overhead for all datagrams.

As for the connection to the WAN, the standard T-1 implementation D1 framing requires less than 1 percent for synchronization, while the more current D4 and Extended SuperFrame (ESF) require only a little additional overhead. Also, where Frame Relay is adopted, its implementation will not exceed the header overheads in X.25. A typical header overhead in Frame Relay consists of only six or less octets. Given these facts, only X.25 protocol was used as the interface protocol to WAN. The protocol applicable within the WAN would have little impact beyond the X.25 interface.

SECTION 5

ANALYSIS MODELING OF FRAME OVERHEAD

5.1 ANALYSIS MODELING

In light of the larger number of protocol characterisitcs and overhead information described in the previous sections, an analysis modeling approach was adopted to determine the effect of protocols on network performance. A set of protocols were selected. It was determined that MHS in the Application layer in OSI (or in its equivalent layer in the Internet) protocol generates the worst frame overhead burden. For similar reasons, the protocol stack selected for the lower layers was the OSI protocol stack. The network environment chosen is that shown in Figure 5-1. The Excel software was used as the modeling tool. The objective of developing the modeling tool was to determine efficiently and accurately all seven layer overhead variations with message size variations or other factors such as layer protocol overhead contributions.

5.2 MODELING ASSUMPTIONS

The following assumptions were made in developing the analysis model. The assumptions are listed by layer.

a. In layer 7, the API overhead can be ignored.

b. All of the overhead import from ASN.1 description of data types and transfer syntax encoding are represented in layer 6.

c. In layer 5, only data service primitives with TPDU is considered, and so one parameter with an assumed length of 254 octets is acceptable. Only two parameters are involved in each session. Each parameter has the maximum allowable size. Concatenation of Session Protocol Data Units (SPDUs) is not allowed.

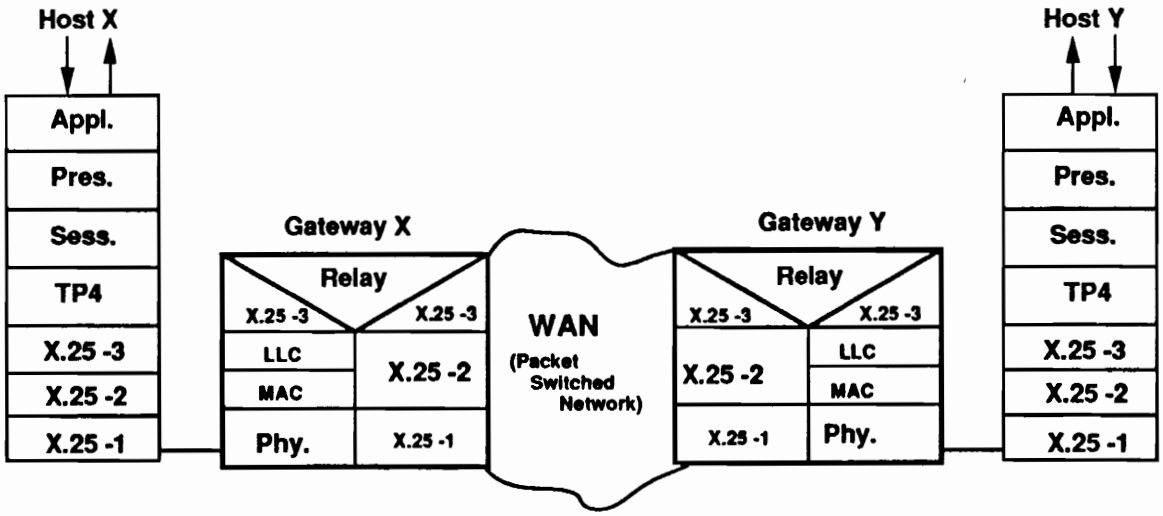


Figure 5-1
Sample OSI Internetworking Environment

d. Class 4 option is applicable in the message transfer mode; this follows the GOSIP version 2 criteria for layer 4. User data is transmitted with the data TPDU only, since the user data is transmitted only after a connection is established. The user data in all other TPDUs is negligible. Maximum TPDU size is established by the layer 3 service. TPDUs are sent sequentially until End-of-Transmission (EOT) is established.

e. The adopted WAN configuration represents a permanent virtual circuit interface without the need for call set up or call clearing. Each subnetwork is harmonized such that ISO 8208 (X.25 Packet Level Protocol for Data Terminal Equipment) will be operating on top of the layer 2 protocols used for access to the subnetwork. Calling and called addresses are each 44 bits long indicating 11 decimal digit addresses. The listing of facilities uses 25 octets only. (These two options do not affect the frame header overhead on data packets.) User data are sent in data packets only. Additonal information field in each control packet is one octet only. Maximum user field length allowed is 128 octets. Six control packets are needed for each connection, however, this does not affect the data packet overhead either.

f. In the data link layer, only the information frames are considered. The maximum size of frames is 1500 octets. Supervisory and unnumbered frames are not included for the header burden determination in the model.

g. The analysis model, although a representation of steady-state conditions, can demonstrate the effect of transmission efficiency loss characteristics especially those related to the data frame header build-up.

5.3 MODEL DESCRIPTION

The analysis model sample is presented in Figure 5-2. The various layers of the OSI protocols are annotated in the leftmost columns. Each layer is described in the PDU Fig

Header Overhead
Analysis for OSI TPDUs

Layer	1 Sublayr Overh'd Bits	2 Data and Layr OH Octets	3 Data and Layr OH Octets	4 Data and Layr OH Octets
7				
Application				
MHS messsage		100	110	121
6				
Presentation				
Address Encoding				
-Overhead		4240	4240	4240
MHS Message				
w/ ASN.1 and encoding OH		4340	4350	4361
5				
Session				
SPDU Fields				
Session Identifier	8			
L Identifier	8			
Parameter Identifier	8			
Par. L Identifier	8			
Parameter Value	32			
Subtotal	64			
Total Layer OH		8	8	8
MHS Message w/OH		4348	4358	4369

Figure 5-2
Sample Analysis Model For Overhead Determination

Header Overhead
Analysis for OSI TPDUs

	1	2	3	4
4				
Transport				
TPDU Type 5				
L Indicator (Max L 254 B)	8			
Fixed part				
TPDU Type (1111)	4			
Credit -flow control	4			
Destination Ref	16			
EOT	1			
TPDU NR (Send Seq. No.)	7			
User Data				
Subtotal	40			
Total Layer OH		5	5	5
MHS Message w/OH		4353	4363	4374
Percentage Increase in Message Size			0	0
3				
Network				
Data Packet Format				
Q (Qualified)	1			
D	1			
Modulo	2			
Group	4			
Channel	8			
Piggyback	3			
More (Grouping)	1			
Sequence	3			
Control	1			
Data				
Subtotal	24			

Figure 5-2 (continued)
Sample Analysis Model For Overhead Determination

Header Overhead
Analysis for OSI TPDUs

	1	2	3	4
Total OH on data packet		3	3	3
Integral Number of 128 Octet Pkts				
from Layer 4 above w/OH		35	35	35
MHS Message with Layer 3 OH		4585	4585	4585
2				
Data Link				
LAP B sublayer				
DSAP/SSAP Address	8			
Control	8			
Data >= 0				
Checksum	16			
Flag	8			
Subtotal	40			
Total OH		5	5	5
Integral number of 131 octet -				
layer 2 packets		36	36	36
Final MHS Data Size for Transmsn		4896	4896	4896
Percentage Increase in Message Size			0	0
Total overhead in this transmission		4796	4786	4775
OH as % of Message		4796	4351	3946

Legend:
EOT = End of Transport
OH = Overhead
Par. = Parameter
PDU = Protocol Data Unit
SAP = Service Access Point

Figure 5-2 (concluded)
Sample Analysis Model For Overhead Determination

components with the associated frame format bits. The standards prescribe the limits of the frames and the partitions within each frame. Where detailed sizing information is not available or the standards do not specify a norm, a conservative estimate is made. For instance, in the Session layer PDU, parameter value is estimated to be 2000 bits. This is very conservative.

The rows on the top of the model format are broken into 18 columns. The first column includes the sublayer framing bits. Each of the following columns contains message octets with or without the header octets as the case may be. An integral number of packets of 128 octet each, including the overhead, is determined in layer 3. Similarly, an integral number of 1500 octet frames, including the overhead, in layer 2 are derived.

Going down the rows, the message or PDU size increases as it picks up the overhead contributed by each layer. Finally, below layer 2, the message with all its header overhead is available for transmission. In the scenario just described, the transmission takes place in one direction only. The message traverses from the WAN down the gateway into the LAN. It could also traverse similarly from a host down to LAN in the other directions.

5.4 ANALYSIS FINDINGS

The enclosed graphs describe the results of the analysis. The variation in the header overhead as a percentage of the message is plotted against the message size variation. Also, the variation in the header overhead size is plotted against the variation in the message size. In the first set of plots, the message size is varied over a wide range - from 100 octets to over 26 Million octets. The very short 100 octet messages could probably be short monologues in an interactive mode while the 2.6 million octetmessages

could be long message file transfers. In the plots following Figures 5-3 through 5-8 , sets of different message sizes are used to project perspectives of different granularity and the consequent effects of overhead on increases in the message size.

The findings from the analysis modeling are very similar to the simulation modeling results. It appears that as the message size increases, the frame header overhead burden becomes less and less important. The performance of the network appears to be dependent on only the message length itself as the message size increase measurably beyond the 1500 octet LAN frame limitation; there is little noticeable increase in the header overhead as a percentage of the message length.

In other words, the throughput of the system underlying the current protocol environment becomes a function of the message or packet size. Greater the message, lesser would be the throughput since the header burden becomes an insignificant factor. This follows a simulation analysis of a bidirectional bridge throughput in a very large network [8] using TCP/IP over Ethernet and a random mix of packet sizes. When message size is very large, the throughput appears to be dependent on the message size only and therefore on the network component's filtering/forwarding capacity and similar other features. The efficiency of the overall network is then controlled by two factors - the throughput of the slowest or least efficient of the transfer system or the delivery system and the overall message size or number of packets of a given size constituting the message. The second set of plots show that the header overhead in absolute octets remain in a predictable range for short messages. However, the overhead build-up follows closely the message size increase pattern.

From the above observations it is also apparent that a further detailed analysis modeling of the non-OSI protocols for this report is not relevant. The TCP/IP

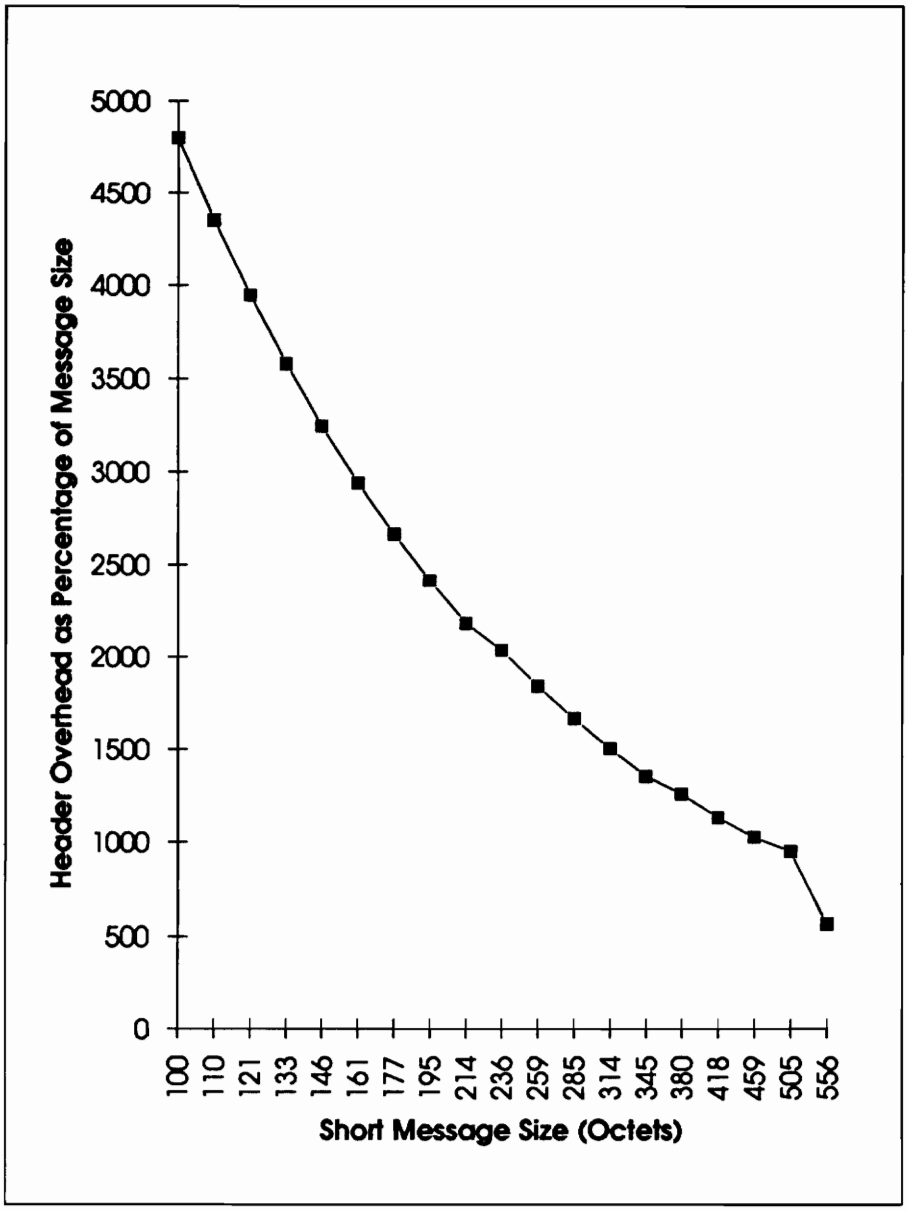


Figure 5-3
Message Versus Overhead Increment

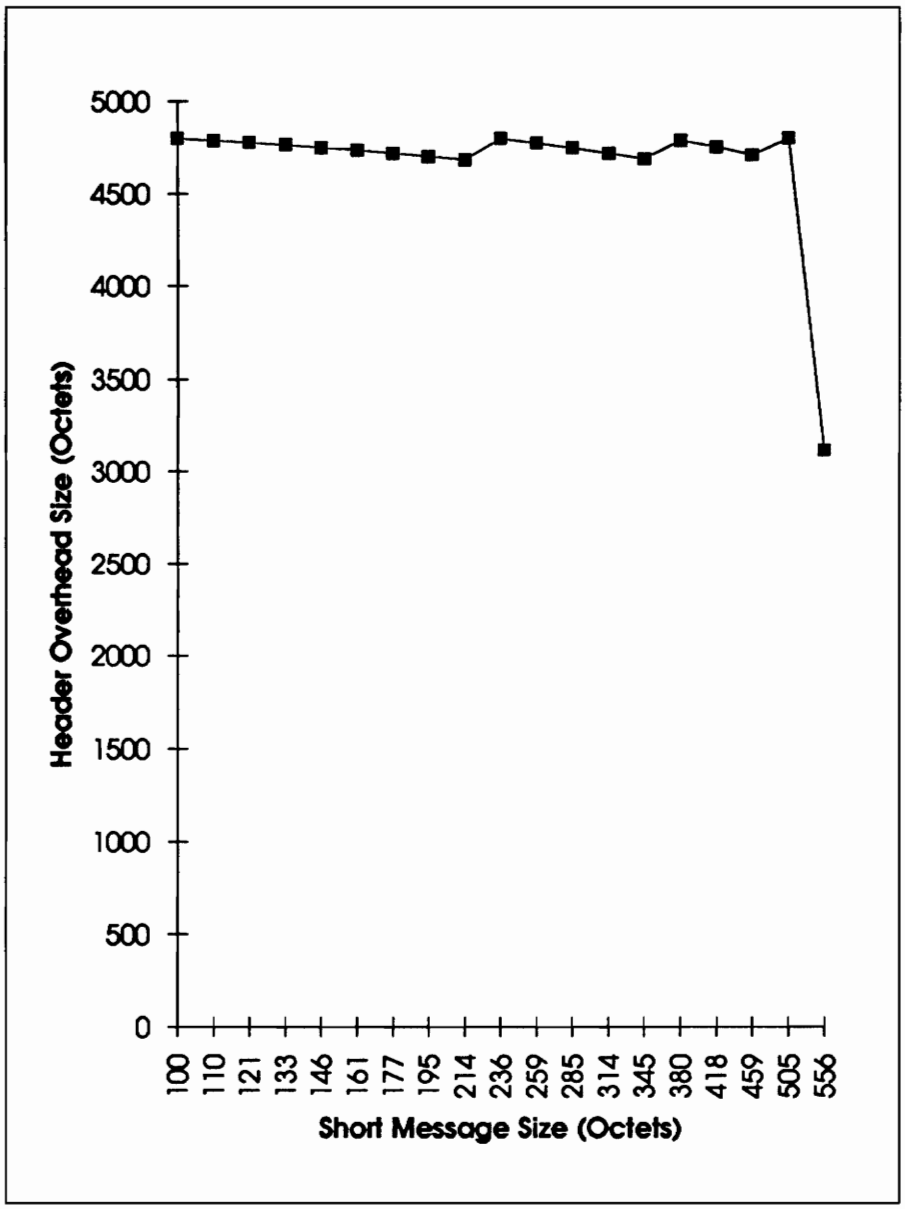


Figure 5-4
Header Variations with Message Size Variations

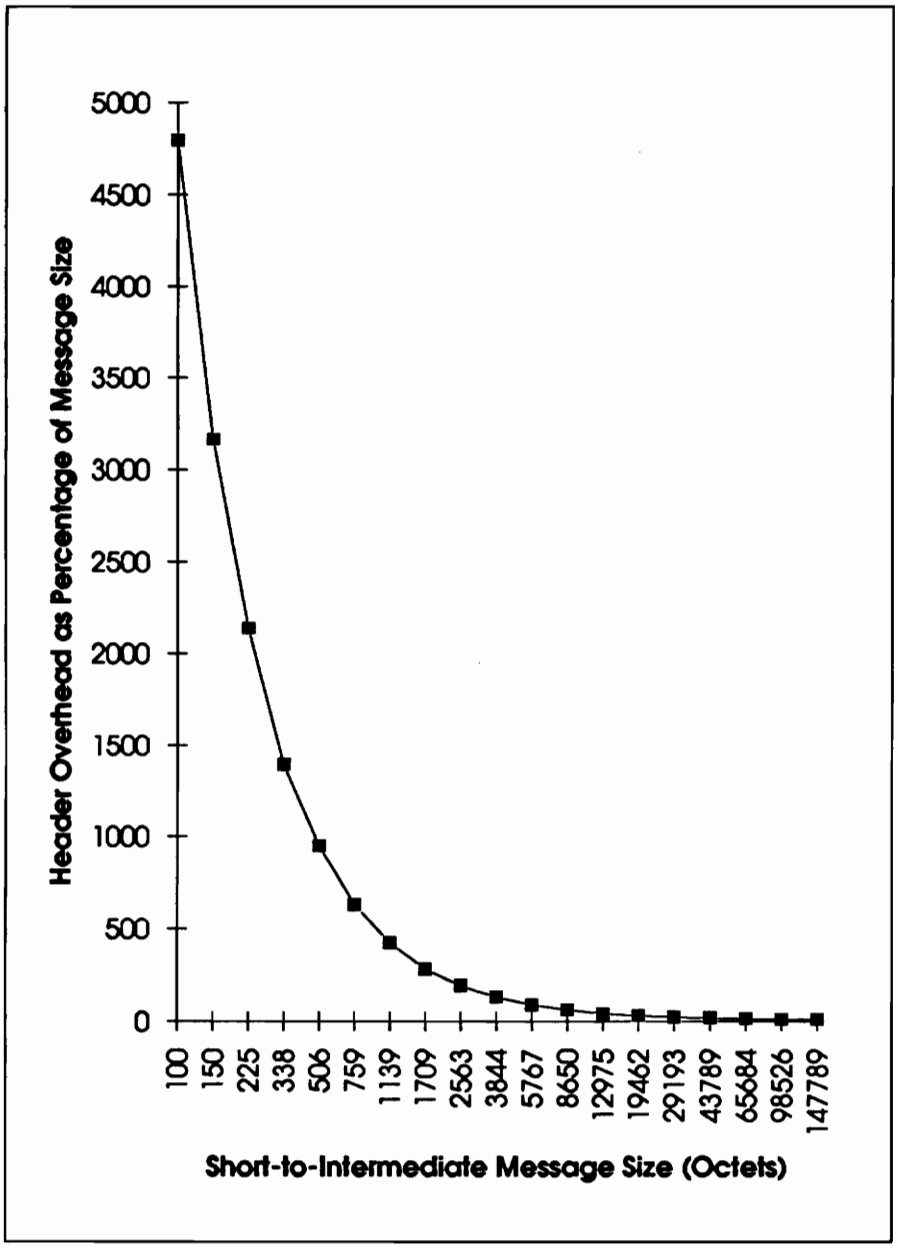


Figure 5-5
Message Versus Overhead Increment

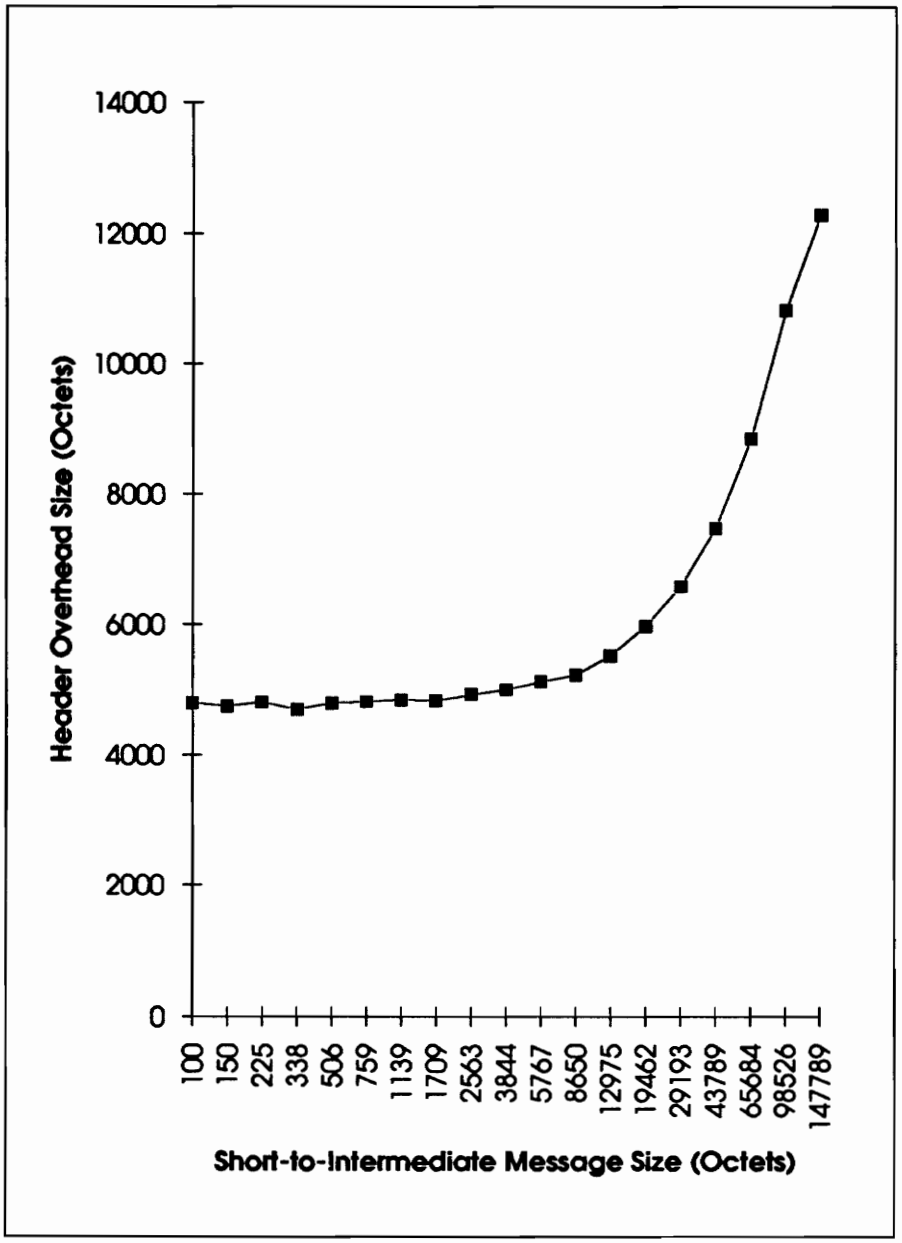


Figure 5-6
Header Variations with Message Size Variations

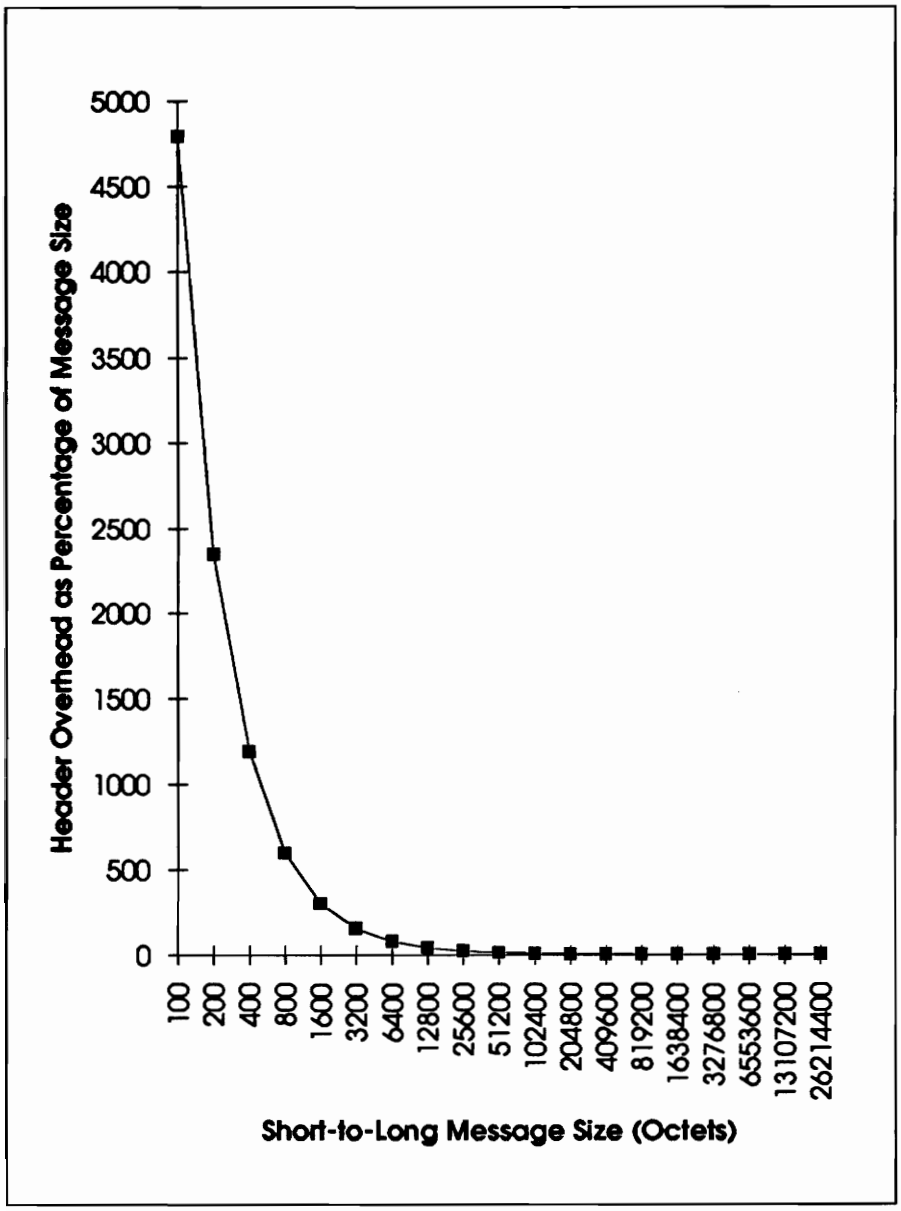


Figure 5-7
Message Versus Overhead Increment

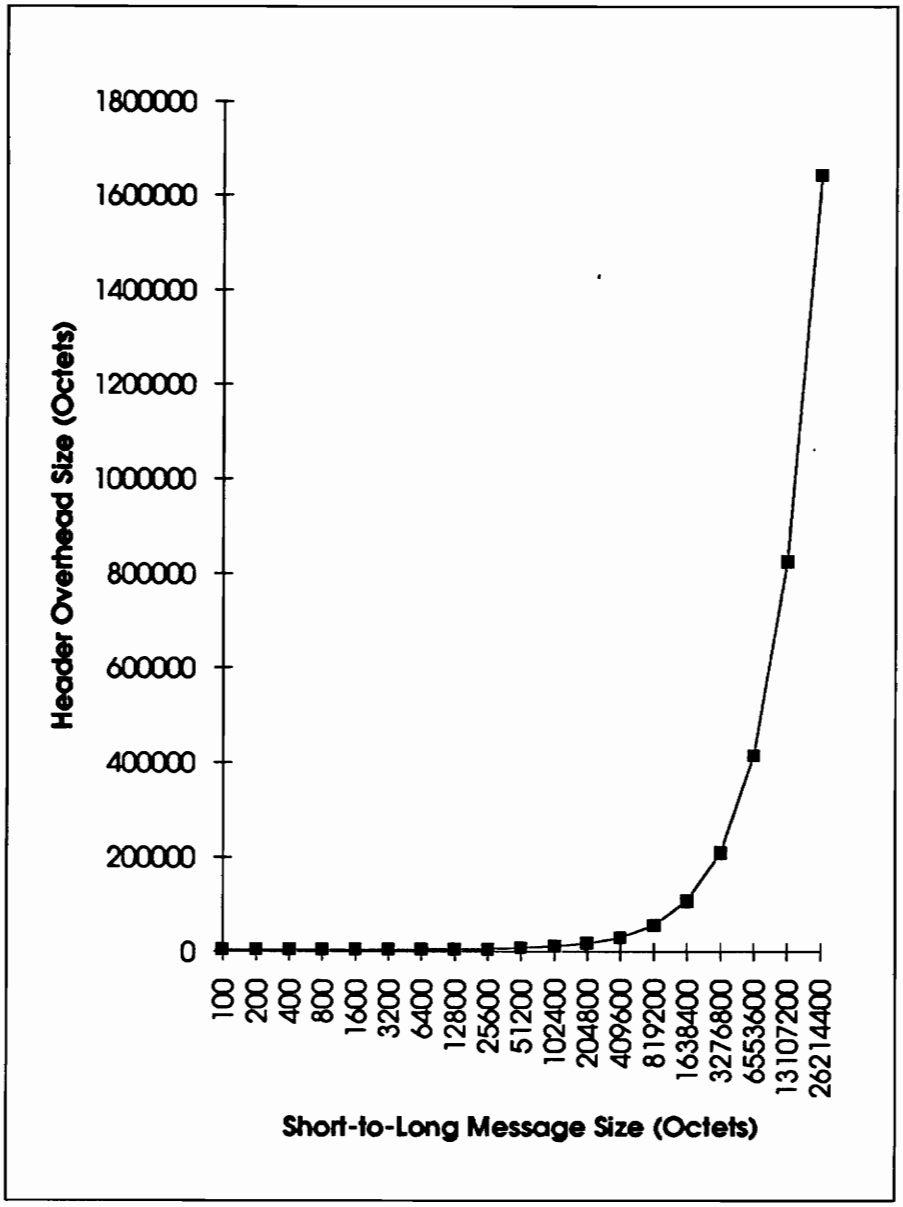


Figure 5-8
Header Variations with Message Size Variations

headercontributions would have little discernible effect beyond a certain message size. A comparative analysis is nevetheless presented in the following concluding section.

SECTION 6

CONCLUSIONS

6.1 FRAME OVERHEAD COMPARISON

In the prior sections, the functional aspects and applicabilities of two groups of protocols for message and file transfer environments, namely, MHS and FTAM, and SMTP and FTP have been discussed. The generic header overhead bounds for the remaining upper and lower layers of the protocol stacks under consideration were outlined. The observations made so far are summarized as follows with pictoral comparisons.

The contribution of headers in the layers of the OSI protocols and Internet protocols are shown in Figures 6-1 and 6-2. The figures demonstrate the header overlay in each of the three categories: SMTP/FTP, MHS and FTAM. As can be expected, the non-OSI protocol set-up will always import the least header overhead. The MHS incurs the most. The overhead assumed was for the case of a single originator and single recipient in all cases. This is because the MHS protocol transgresses the presentation layer in the OSI protocol stack. The presentation and session layers' combined overhead is considered to be 40 octets and shown both for the MHS and FTAM protocols. A 600 octet overhead for MHS alone should be reasonable.

The overhead advantage increases significantly if the data messages do not need to be fragmented too small. The FTAM also has a low header overhead once it is set-up and is in an ongoing mode. In some cases, the ongoing header overhead in FTAM may be less than the FTP if FTAM is tailored with optimized number of options and a compressed addressing scheme.

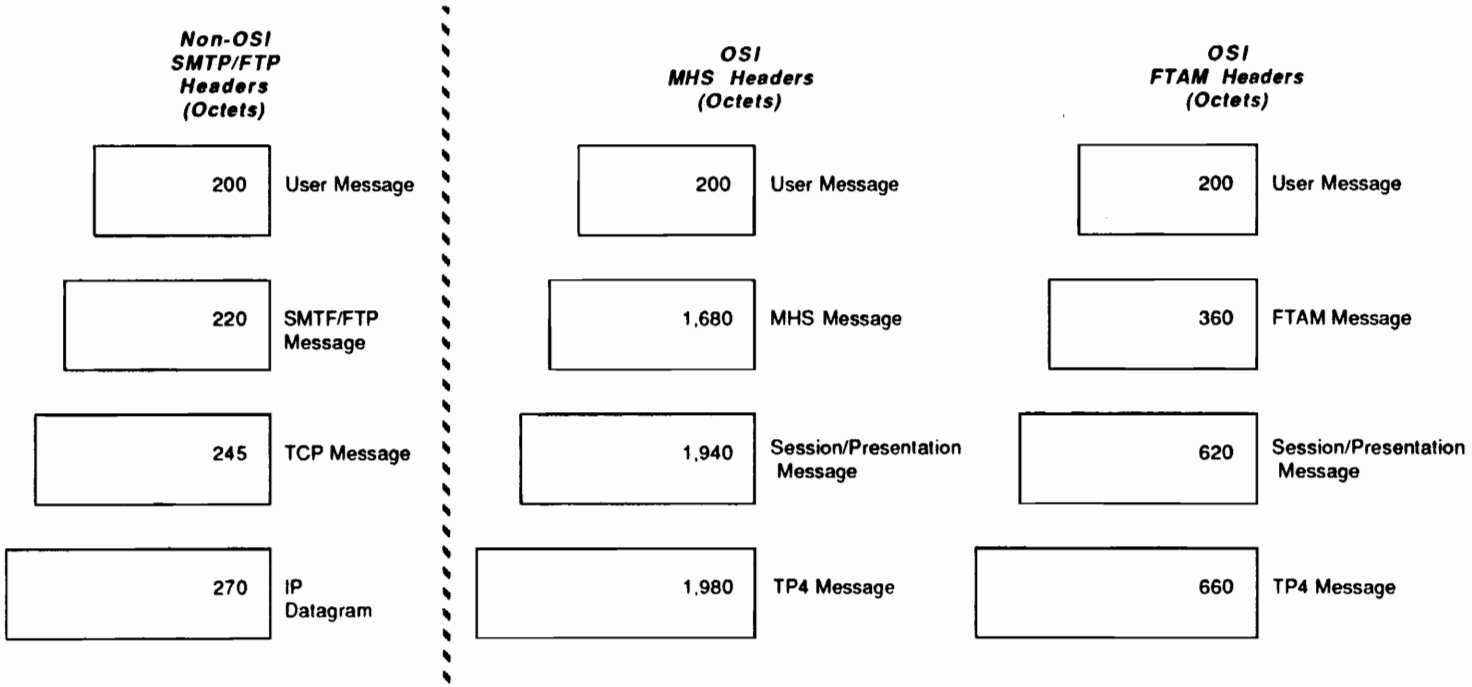


Figure 6-1
Representative Header Comparison for a 200 Octet User Message

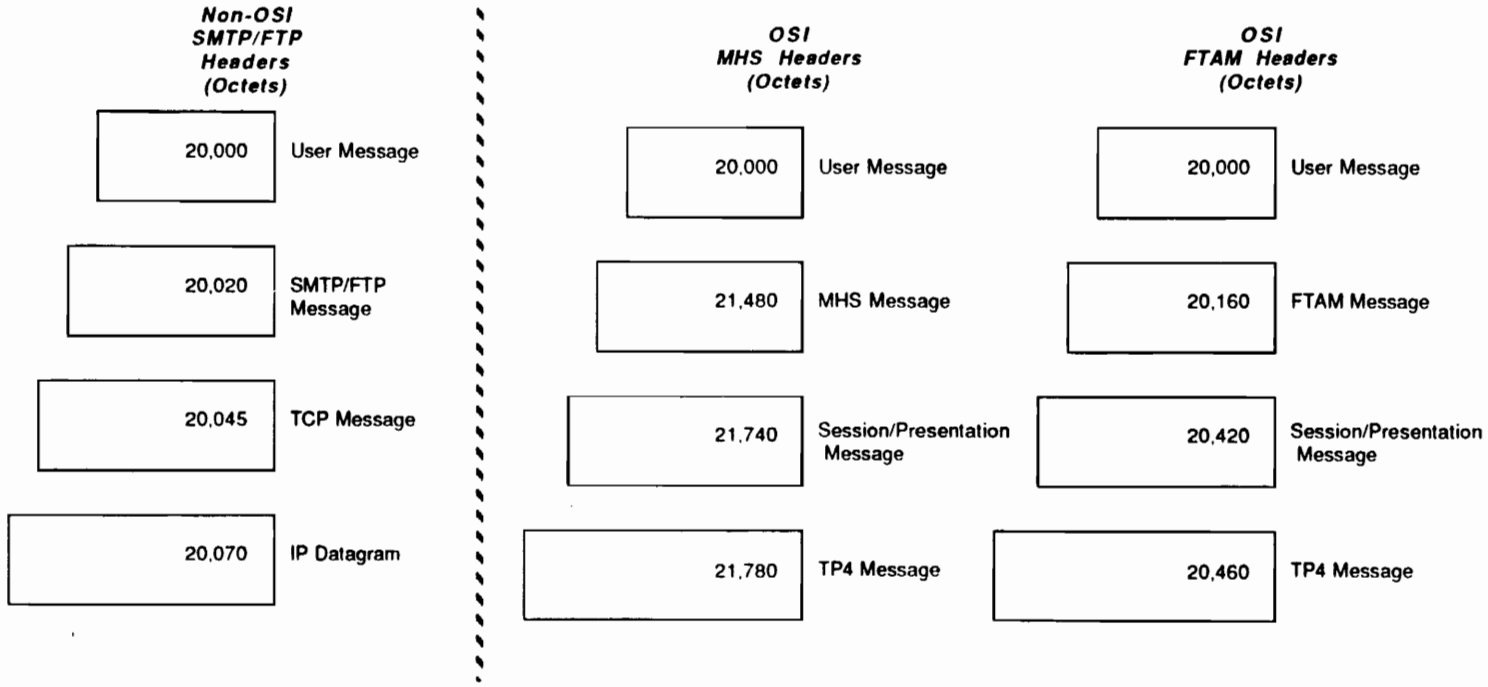


Figure 6-2
Representative Header Comparison for a 20,000 Octet User Message

In Figure 6-1, a 200 octet message is selected to represent a short transaction message. This message is shown in the top box in this category of protocol as well in other procotols. In the subsequent boxes below the user message, the applicable layer header overhead is added as per the criteria discussed in Sections 2, 3 and 4. When all layer overheads are added, we arrive at the bottom box which shows the final message size at layer 4. It is seen that MHS and FTAM represent the largest and the second largest messages respectively.

In Figure 6-2, the same concept of importing header overheads is applied to a much larger message size of 20,000 octets. The layer 4 message size clearly indicates that in this protocol scenario, the LAN 1500 octets limit is quickly exceeded.

Thus, transfer of large files as well as short messages is determined by the limitations imposed by the maximum message size allowed by the delivery system. The processing in the end systems being a function of the upper layer data overhead burden, and processing in the intermediate systems (e.g., routers in between end systems) being a function of the lower layers, it is important that the protocols selected and their implementation incur the minimum message overhead. Finally, it must be pointed out that the processing efficiency of the data fields is also a function of the architecture of the end and intermeadiate systems. For this analysis, it is implicitly assumed that this efficiency is common to all portocols and thus not a variable to be incorporated in the calculations.

6.2 PROTOCOL OVERHEAD OPTIMIZATIONS

In the detailed MHS overhead analysis, it was found that there was a significant overhead associated with the addresses, especially, the ORAddresses. If the addressing

scheme is carefully tailored then in the MHS as well as in the case of other protocols the header overhead can be substantially reduced.

The best address choice would probably be a numeric string, as against the popular "Mnemonic" form, since a large number of addresses can be accomodated in a short one to two octet field. Also, addressing by the organization name, rather than the indiviual originator or recipient name saves some amount of header space. It is also seen that by convention, the ORAddress is repeated twice, once in the message header and once in the message envelope. It may also be repeated in the message content. The information on the envelope is available to the UA. If the UA can be programmed to be aware of this, then there may not be any need for the message or the content to have the ORAddress.

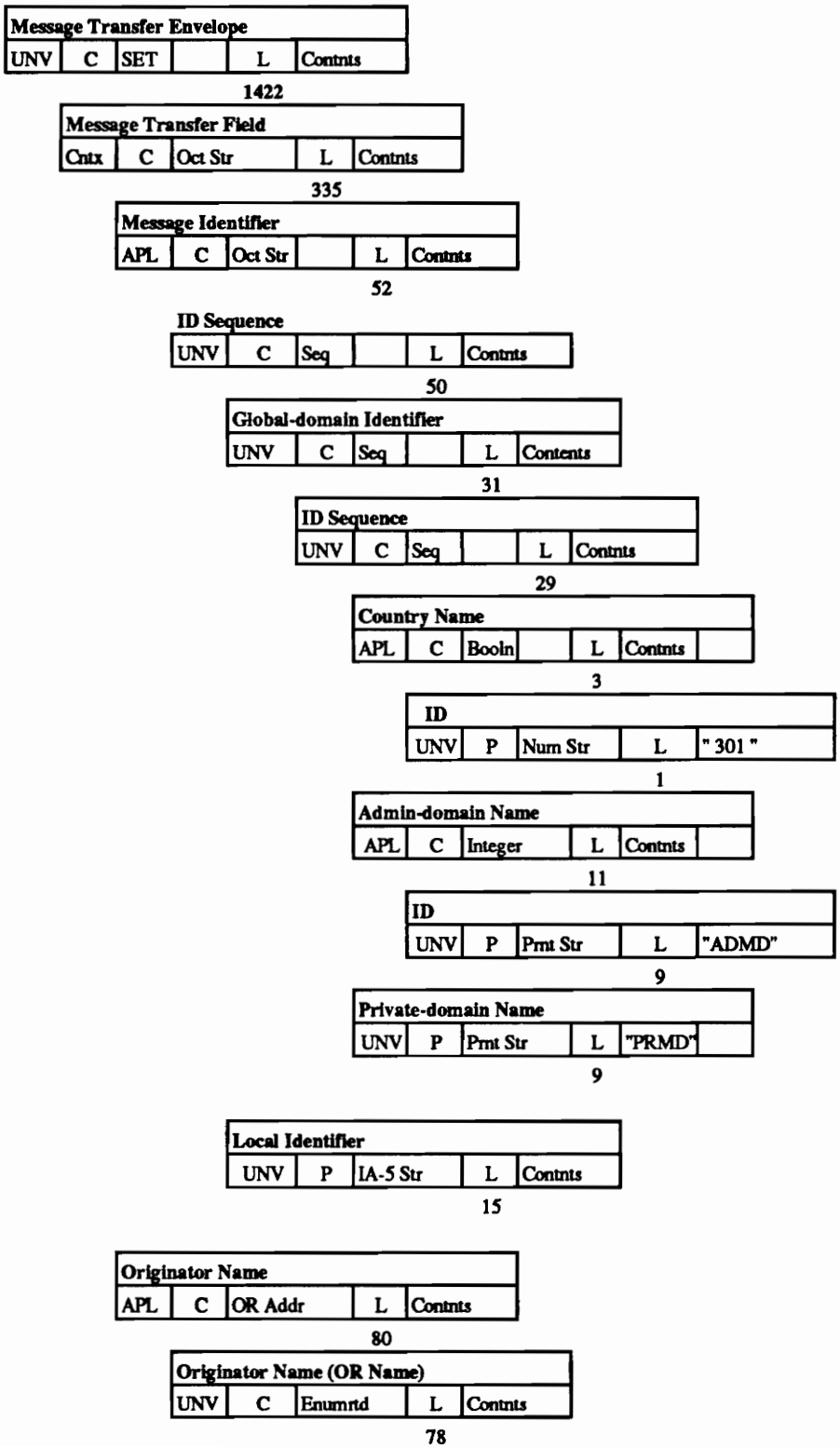
The use of options such as Extension and Trace Information can add substantial overhead to the MHS header field. Additionally, the security features, copy to other recipients add to the encoding and processing load. These features can be somewhat mitigated by alternative security and routing options adopted below layer 3. However, the resulting saving may not be conducive to such adoptions.

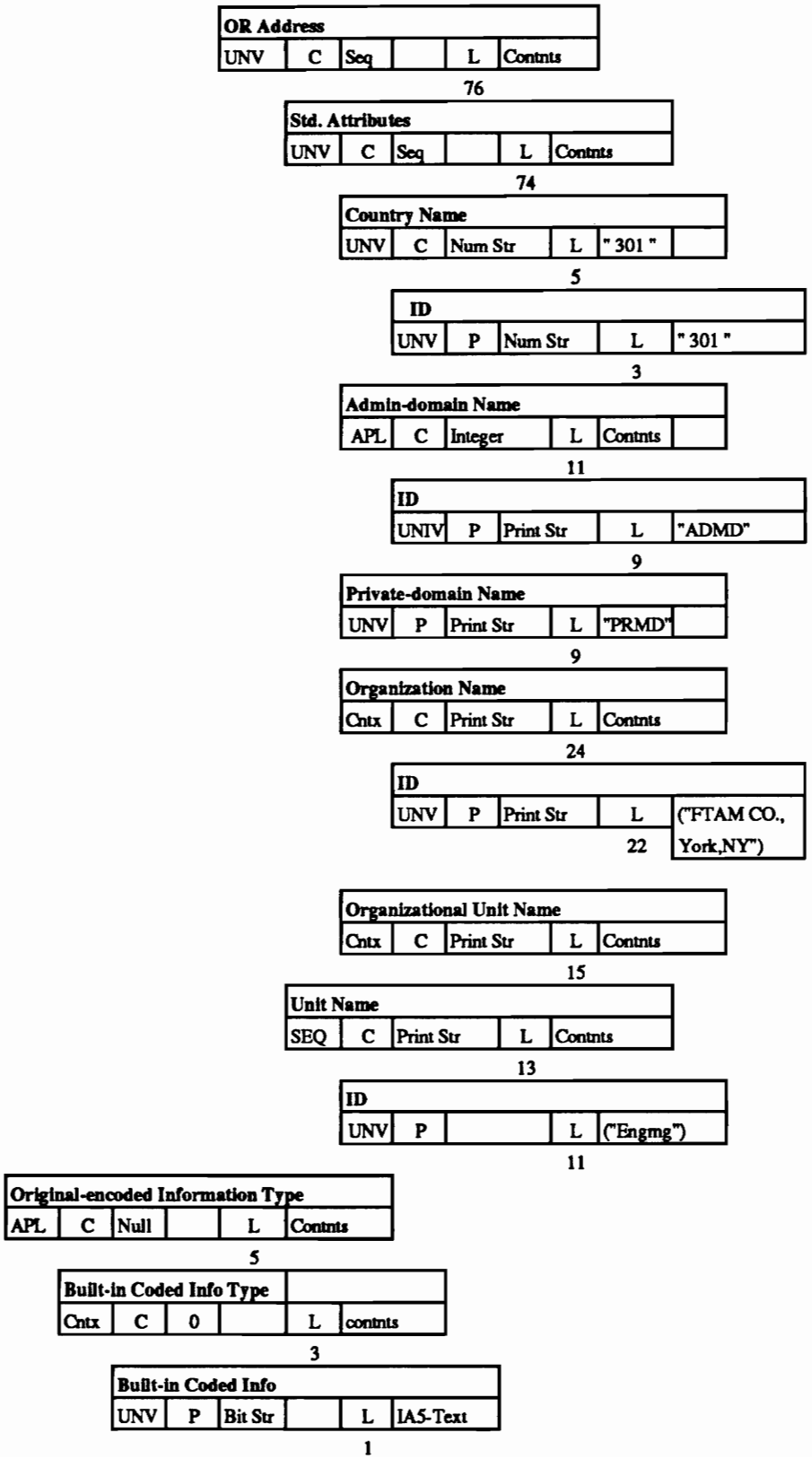
The optimization observations made above with regard to MHS are relevant to a large extent in other protocol environments. In case of the non-OSI protocols, the overhead is much smaller all through layer 4. This is true on not only of Internet but other proprietary protocols in general. In the OSI environment, the other major overhead concerns arise out of duplication of functions between layers and redundancy of primitives among several layers. The part of the reason is the requirement to offer a degree of flexibility to the users and the other reason being the tendency to satisfy all and sundry toward *universal* compatibility. It is believed that with time as OSI protocols spread across all user groups, the protocol primitives will become more compact and

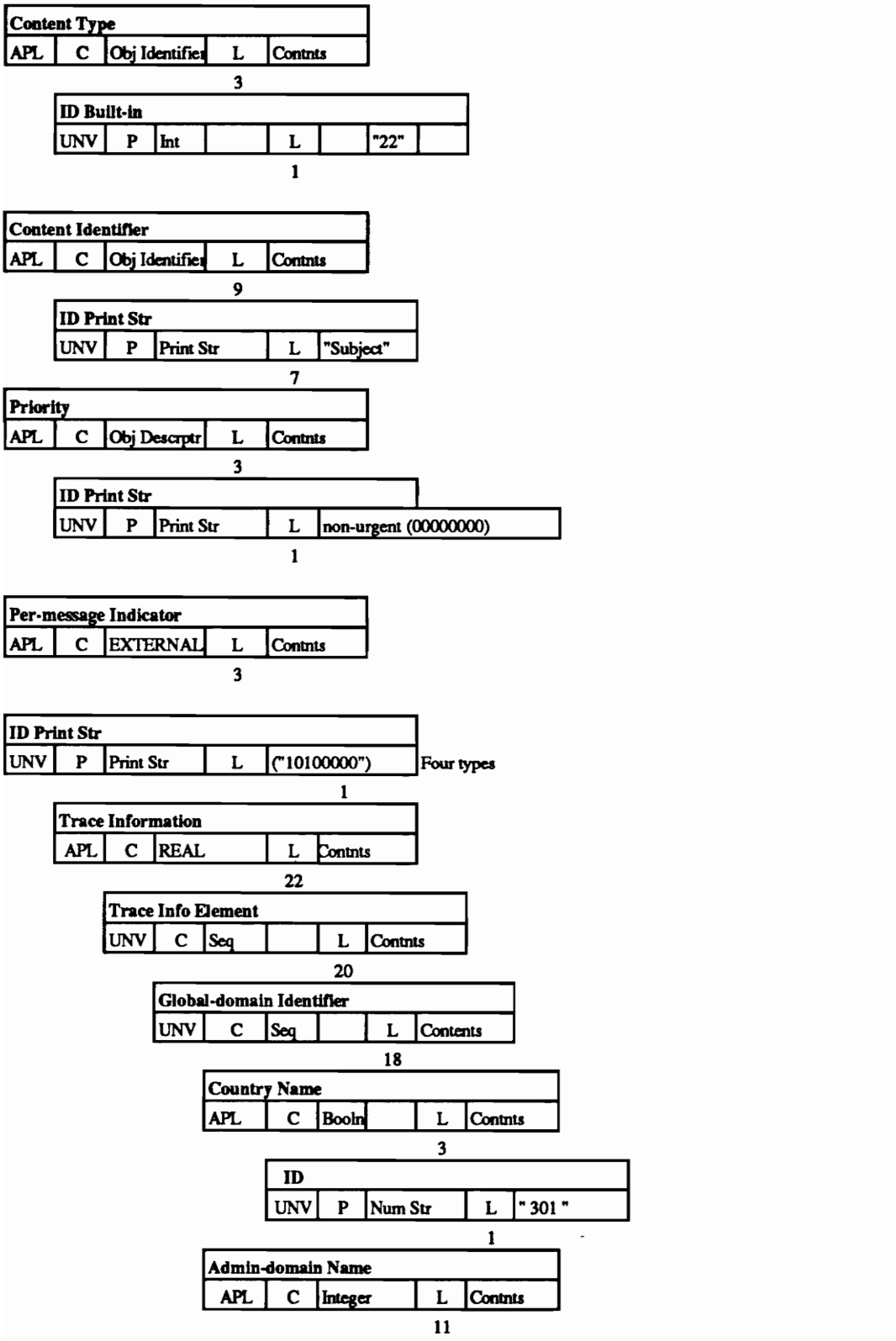
focussed to the needs of the end user.

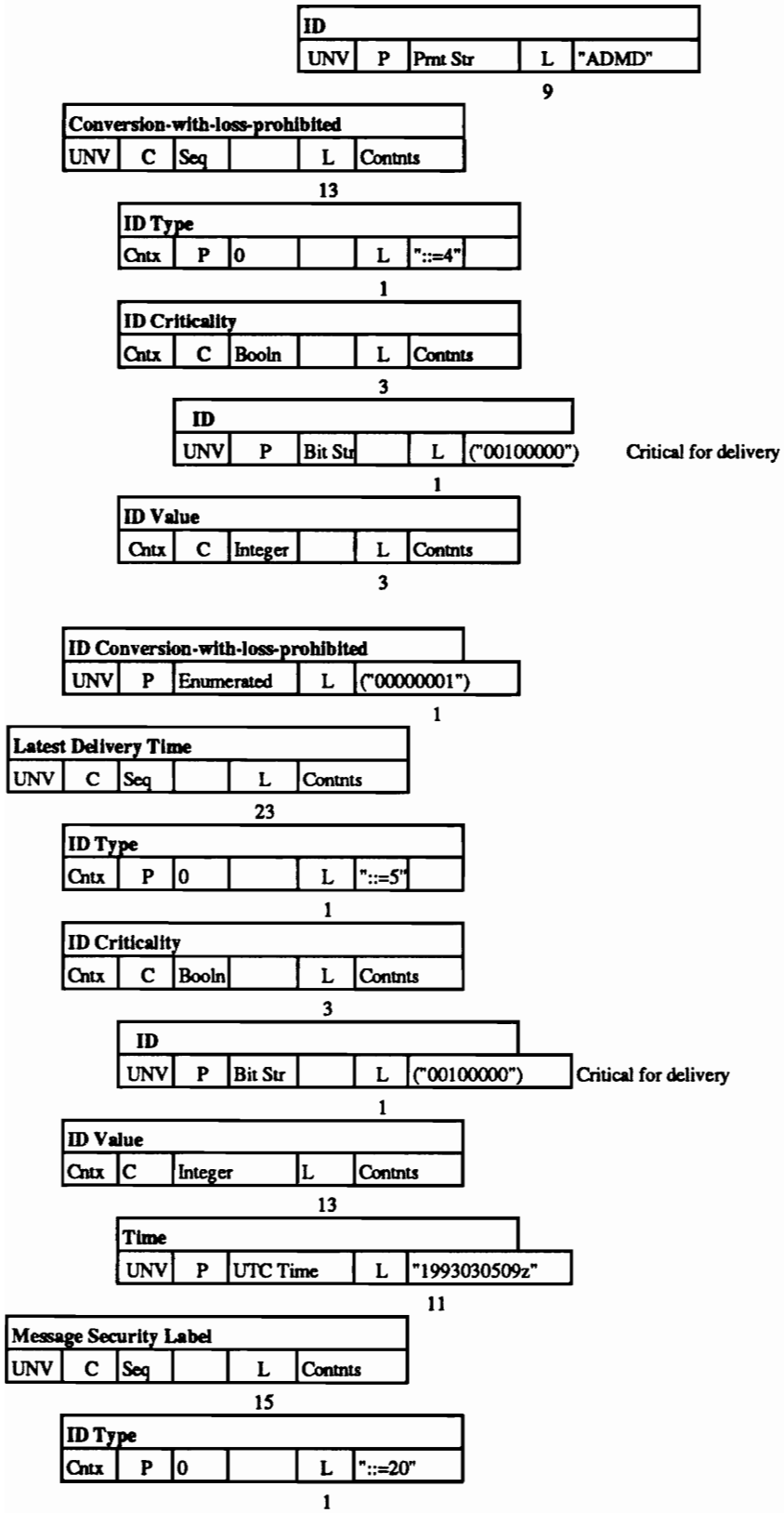
APPENDIX A

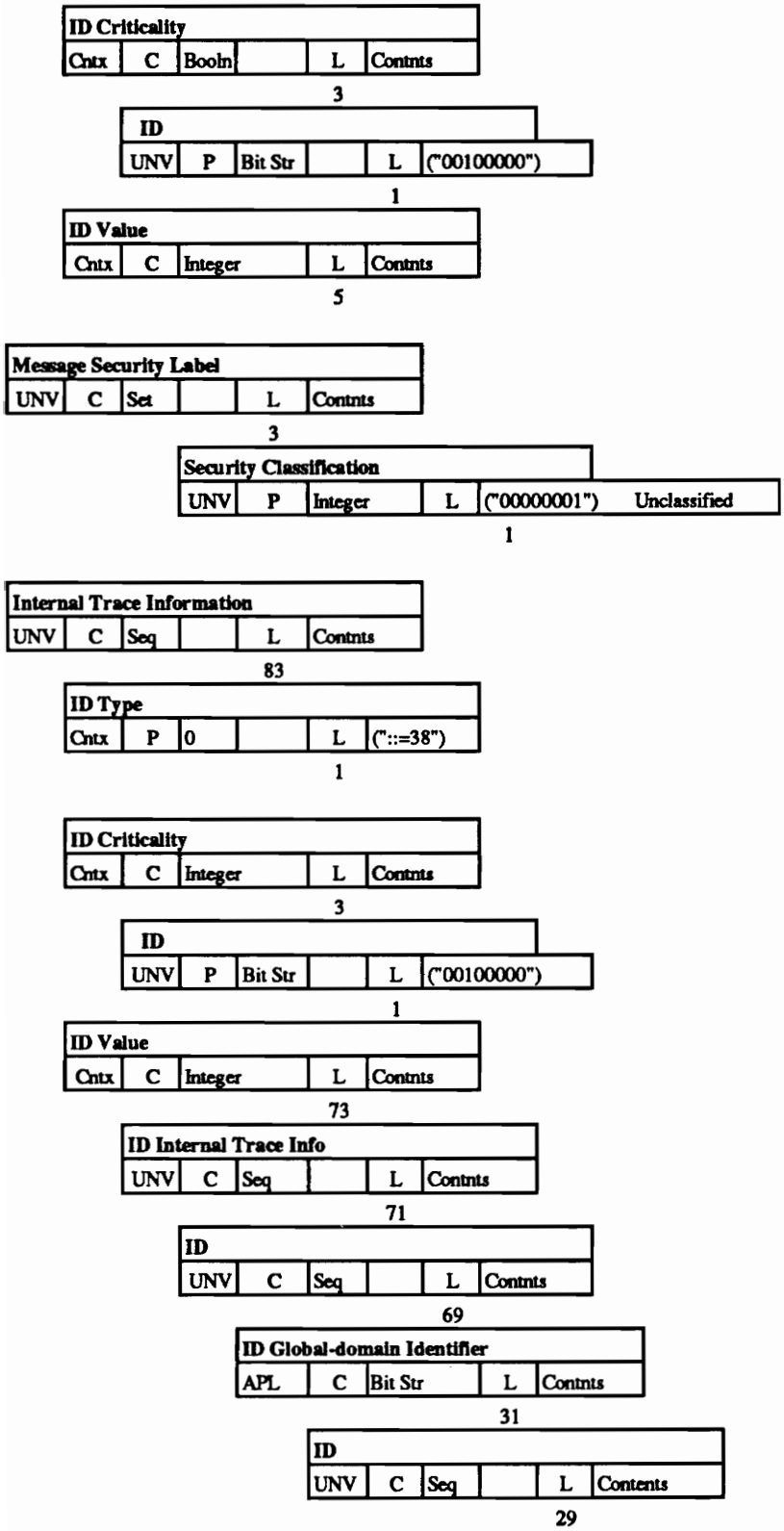
MESSAGE TRANSFER ENVELOPE HEADER ENCODING

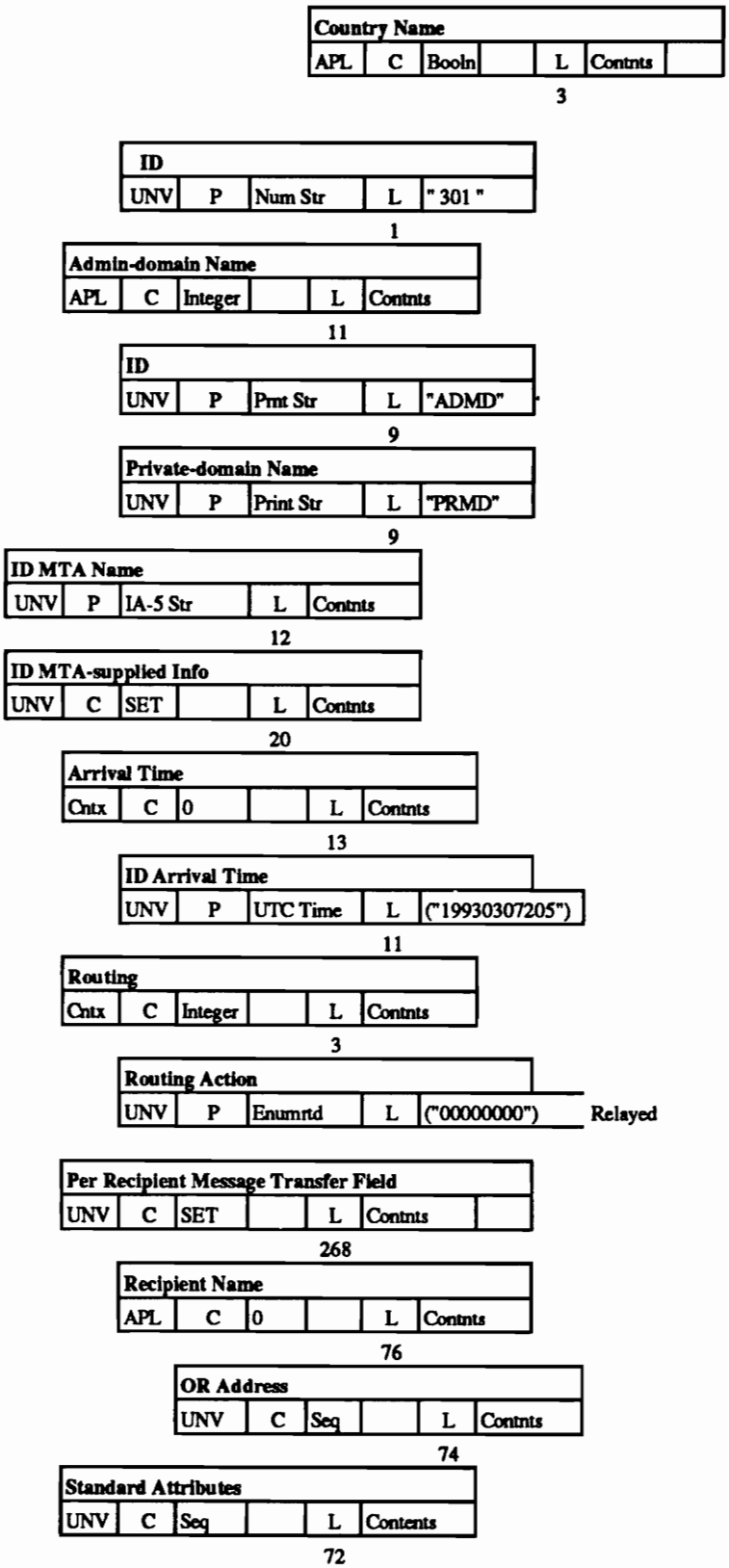

Message Transfer Envelope
UNV C SET L Contnts
1422
Message Transfer Field
Cntx C Oct Str L Contnts
335
Message Identifier
APL C Oct Str L Contnts
52
ID Sequence
UNV C Seq L Contnts
50
Global-domain Identifier
UNV C Seq L Contents
31
ID Sequence
UNV C Seq L Contnts
29
Country Name
APL C Booln L Contnts
3
ID
UNV P Num Str L " 301 "
1
Admin-domain Name
APL C Integer L Contnts
11
ID
UNV P Prnt Str L "ADMD"
9
Private-domain Name
UNV P Prnt Str L "PRMD"
9
Local Identifier
UNV P IA-5 Str L Contnts
15
Originator Name
APL C OR Addr L Contnts
80
Originator Name (OR Name)
UNV C Enumrtd L Contnts
78

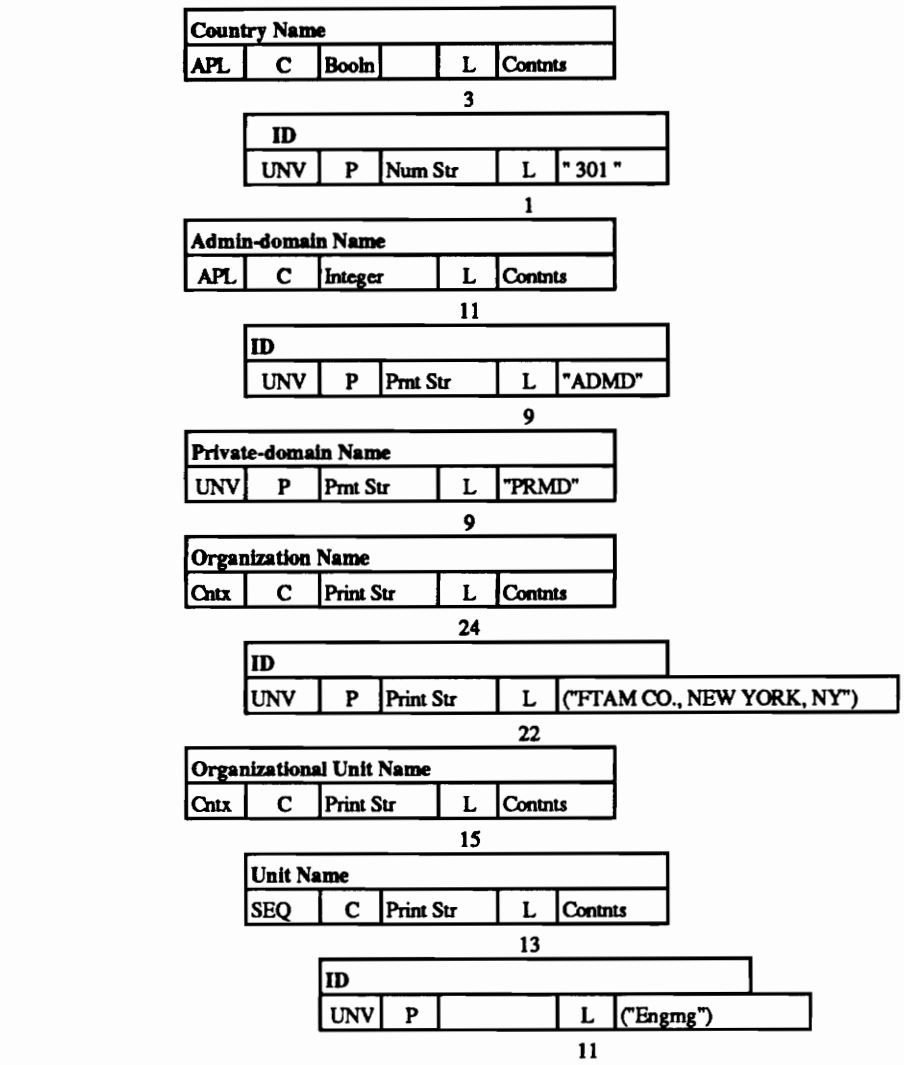

OR Address
UNV C Seq L Contnts
76
Std. Attributes
UNV C Seq L Contnts
74
Country Name
UNV C Num Str L " 301 "
5
ID
UNV P Num Str L " 301 "
3
Admin-domain Name
APL C Integer L Contnts
11
ID
UNIV P Print Str L "ADMD"
9
Private-domain Name
UNV P Print Str L "PRMD"
9
Organization Name
Cntx C Print Str L Contnts
24
ID
UNV P Print Str L ("FTAM CO., York,NY")
22
Organizational Unit Name
Cntx C Print Str L Contnts
15
Unit Name
SEQ C Print Str L Contnts
13
ID
UNV P L ("Engmg")
11
Original-encoded Information Type
APL C Null L Contnts
5
Built-in Coded Info Type
Cntx C 0 L contnts
3
Built-in Coded Info
UNV P Bit Str L IA5-Text
1

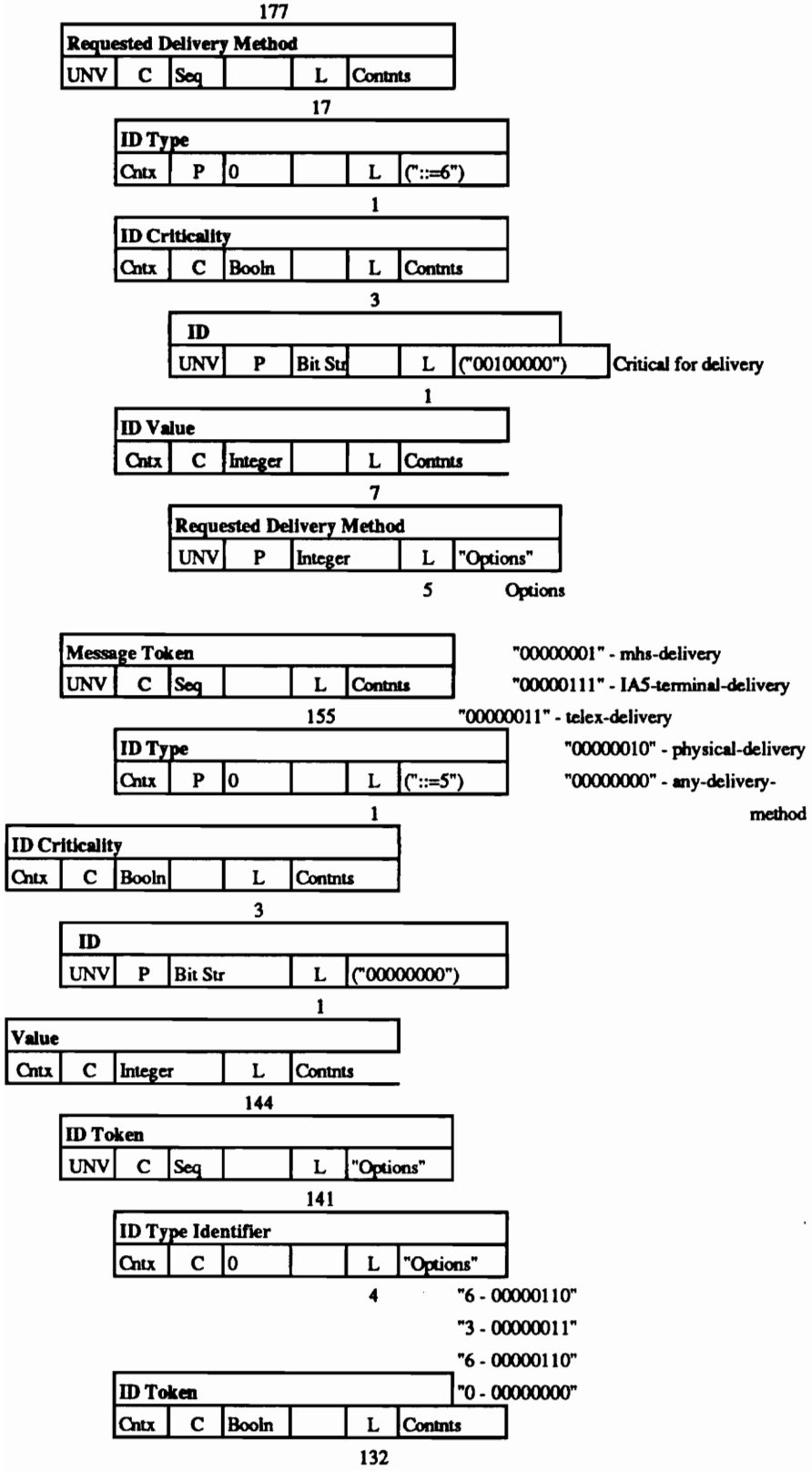

Content Type
APL C Obj Identifier L Contnts
3
ID Built-in
UNV P Int L "22"
1
Content Identifier
APL C Obj Identifier L Contnts
9
ID Print Str
UNV P Print Str L "Subject"
7
Priority
APL C Obj Descrptr L Contnts
3
ID Print Str
UNV P Print Str L non-urgent (00000000)
1
Per-message Indicator
APL C EXTERNAL L Contnts
3
ID Print Str
UNV P Print Str L ("10100000")
Four types
1
Trace Information
APL C REAL L Contnts
22
Trace Info Element
UNV C Seq L Contnts
20
Global-domain Identifier
UNV C Seq L Contents
18
Country Name
APL C Booln L Contnts
3
ID
UNV P Num Str L " 301 "
1
Admin-domain Name
APL C Integer L Contnts
11

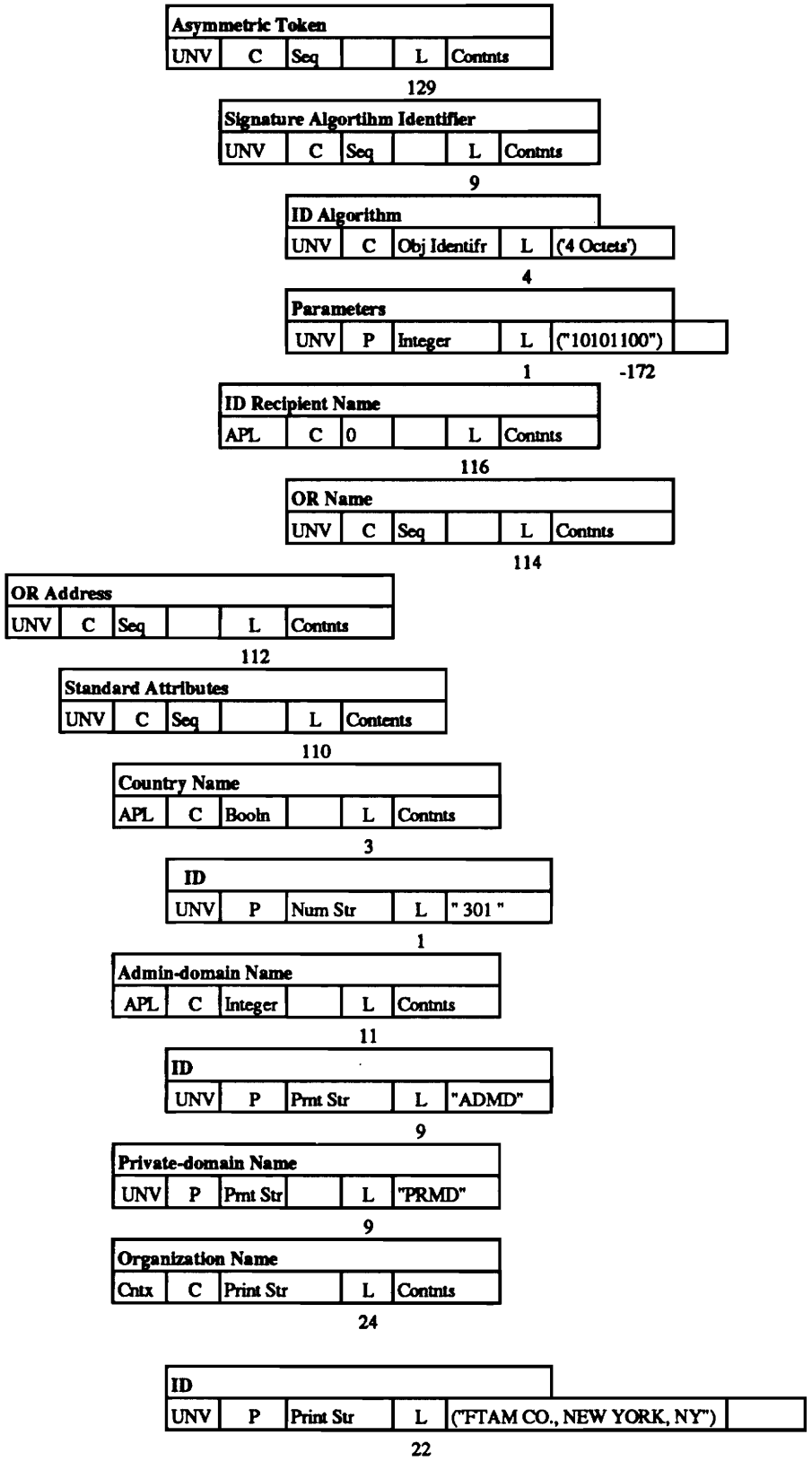

ID
UNV P Prnt Str L "ADMD"
9
Conversion-with-loss-prohibited
UNV C Seq L Contnts
13
ID Type
Cntx P 0 L "::=4"
1
ID Criticality
Cntx C Booln L Contnts
3
ID
UNV P Bit Str L ("00100000")
Critical for delivery
1
ID Value
Cntx C Integer L Contnts
3
ID Conversion-with-loss-prohibited
UNV P Enumerated L ("00000001")
1
Latest Delivery Time
UNV C Seq L Contnts
23
ID Type
Cntx P 0 L "::=5"
1
ID Criticality
Cntx C Booln L Contnts
3
ID
UNV P Bit Str L ("00100000")
Critical for delivery
1
ID Value
Cntx C Integer L Contnts
13
Time
UNV P UTC Time L "1993030509z"
11
Message Security Label
UNV C Seq L Contnts
15
ID Type
Cntx P 0 L "::=20"
1

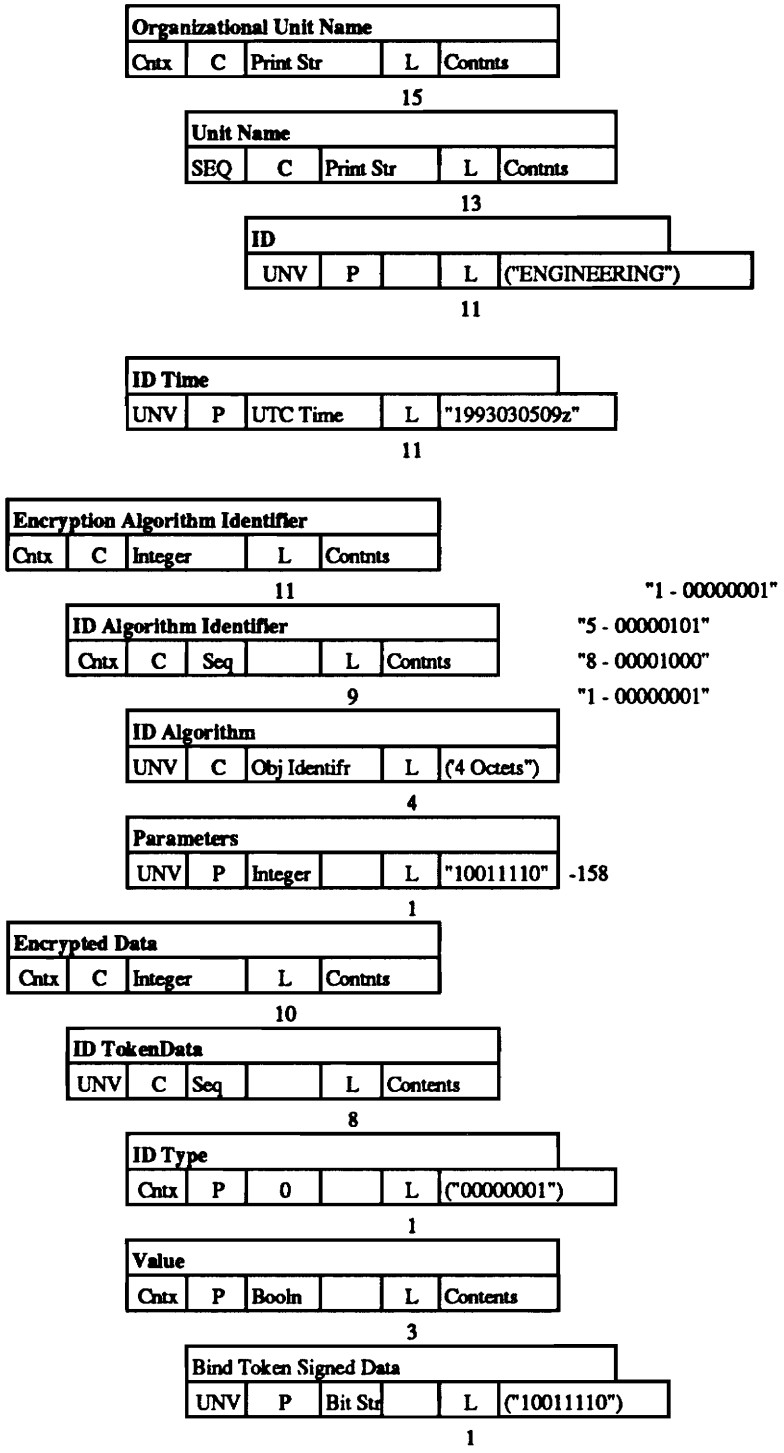

ID Criticality
Cntx C Booln L Contnts
3
ID
UNV P Bit Str L ("00100000")
1
ID Value
Cntx C Integer L Contnts
5
Message Security Label
UNV C Set L Contnts
3
Security Classification
UNV P Integer L ("00000001") Unclassified
1
Internal Trace Information
UNV C Seq L Contnts
83
ID Type
Cntx P 0 L ("::=38")
1
ID Criticality
Cntx C Integer L Contnts
3
ID
UNV P Bit Str L ("00100000")
1
ID Value
Cntx C Integer L Contnts
73
ID Internal Trace Info
UNV C Seq L Contnts
71
ID
UNV C Seq L Contnts
69
ID Global-domain Identifier
APL C Bit Str L Contnts
31
ID
UNV C Seq L Contents
29


Country Name
APL C Booln L Contnts
3
ID
UNV P Num Str L " 301 "
1
Admin-domain Name
APL C Integer L Contnts
11
ID
UNV P Prnt Str L "ADMD"
9
Private-domain Name
UNV P Print Str L "PRMD"
9
ID MTA Name
UNV P IA-5 Str L Contnts
12
ID MTA-supplied Info
UNV C SET L Contnts
20
Arrival Time
Cntx C 0 L Contnts
13
ID Arrival Time
UNV P UTC Time L ("19930307205")
11
Routing
Cntx C Integer L Contnts
3
Routing Action
UNV P Enumrtd L ("00000000") Relayed
1
Per Recipient Message Transfer Field
UNV C SET L Contnts
268
Recipient Name
APL C 0 L Contnts
76
OR Address
UNV C Seq L Contnts
74
Standard Attributes
UNV C Seq L Contents
72


Country Name
APL C Booln L Contnts
3
ID
UNV P Num Str L " 301 "
1
Admin-domain Name
APL C Integer L Contnts
11
ID
UNV P Prnt Str L "ADMD"
9
Private-domain Name
UNV P Prnt Str L "PRMD"
9
Organization Name
Cntx C Print Str L Contnts
24
ID
UNV P Print Str L ("FTAM CO., NEW YORK, NY")
22
Organizational Unit Name
Cntx C Print Str L Contnts
15
Unit Name
SEQ C Print Str L Contnts
13
ID
UNV P L ("Engrng")
11
Originally Specified Recipient
Cntx C 0 L Contnts
3
ID Originally Specified Name
UNV P Integer L ("00000001")
1
Recipient Indicator
Cntx C Booln L Contnts
3
ID Per Recipient Indicator
UNV P Bit Str L ("10101000")
1
Extensions Field
Cntx C Seq L Contnts


177
Requested Delivery Method
UNV C Seq L Contnts
17
ID Type
Cntx P 0 L ("::=6")
1
ID Criticality
Cntx C Booln L Contnts
3
ID
UNV P Bit Str L ("00100000")
Critical for delivery
1
ID Value
Cntx C Integer L Contnts
7
Requested Delivery Method
UNV P Integer L "Options"
5
Options
Message Token
UNV C Seq L Contnts
155
ID Type
Cntx P 0 L ("::=5")
1
ID Criticality
Cntx C Booln L Contnts
3
ID
UNV P Bit Str L ("00000000")
1
Value
Cntx C Integer L Contnts
144
ID Token
UNV C Seq L "Options"
141
ID Type Identifier
Cntx C 0 L "Options"
4
ID Token
Cntx C Booln L Contnts
132
"00000001" - mhs-delivery
"00000111" - IA5-terminal-delivery
"00000011" - telex-delivery
"00000010" - physical-delivery
"00000000" - any-delivery-method
"6 - 00000110"
"3 - 00000011"
"6 - 00000110"
"0 - 00000000"


Asymmetric Token
UNV C Seq L Contnts
129
Signature Algortihm Identifier
UNV C Seq L Contnts
9
ID Algorithm
UNV C Obj Identifr L ('4 Octets')
4
Parameters
UNV P Integer L ("10101100")
1
-172
ID Recipient Name
APL C 0 L Contnts
116
OR Name
UNV C Seq L Contnts
114
OR Address
UNV C Seq L Contnts
112
Standard Attributes
UNV C Seq L Contents
110
Country Name
APL C Booln L Contnts
3
ID
UNV P Num Str L " 301 "
1
Admin-domain Name
APL C Integer L Contnts
11
ID
UNV P Prnt Str L "ADMD"
9
Private-domain Name
UNV P Prnt Str L "PRMD"
9
Organization Name
Cntx C Print Str L Contnts
24
ID
UNV P Print Str L ("FTAM CO., NEW YORK, NY")
22

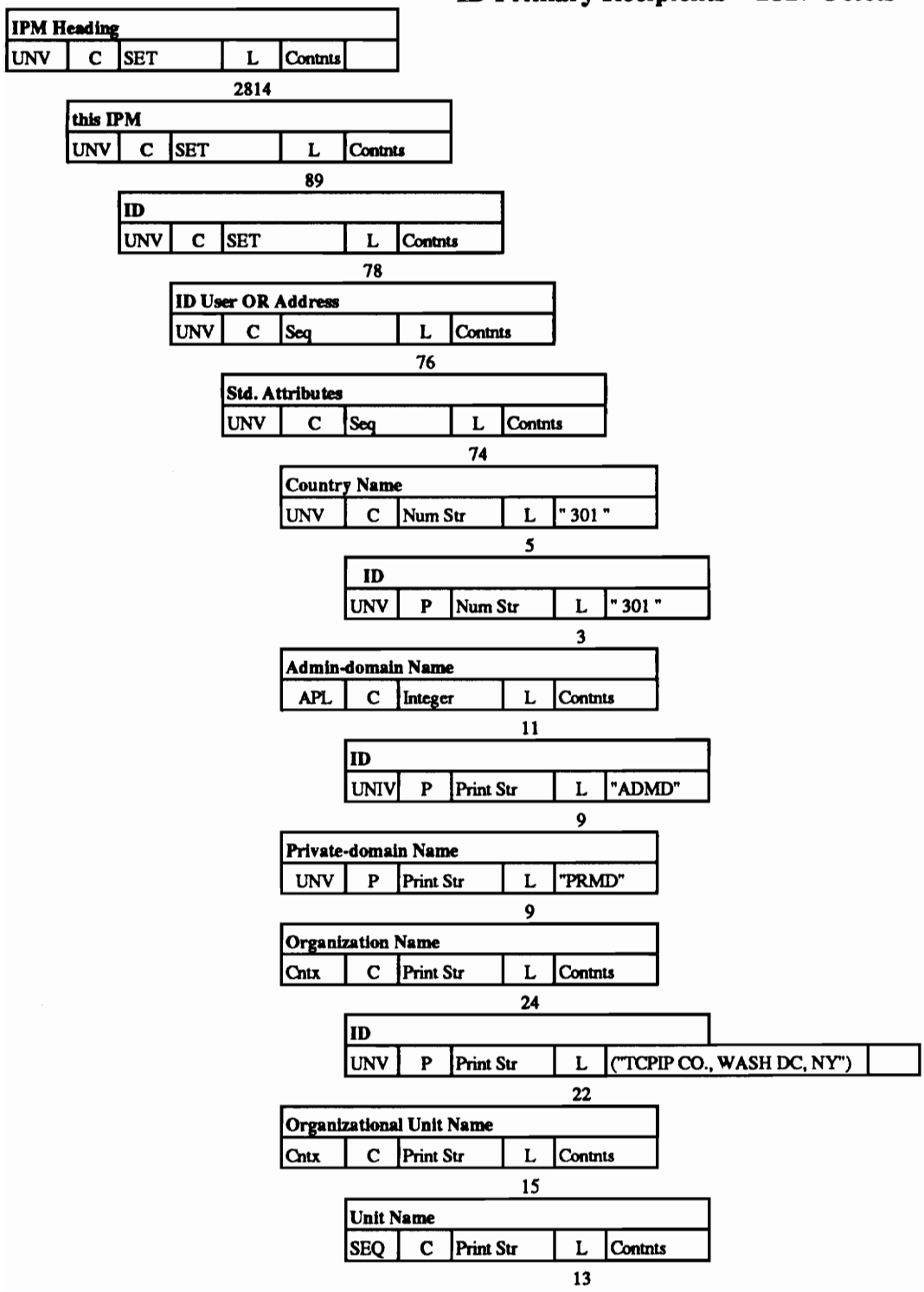

Organizational Unit Name
Cntx C Print Str L Contnts
15
Unit Name
SEQ C Print Str L Contnts
13
ID
UNV P L ("ENGINEERING")
11
ID Time
UNV P UTC Time L "1993030509z"
11
Encryption Algorithm Identifier
Cntx C Integer L Contnts
11
"1 - 00000001"
ID Algorithm Identifier
Cntx C Seq L Contnts
"5 - 00000101"
"8 - 00001000"
9
"1 - 00000001"
ID Algorithm
UNV C Obj Identifr L ("4 Octets")
4
Parameters
UNV P Integer L "10011110"
-158
1
Encrypted Data
Cntx C Integer L Contnts
10
ID TokenData
UNV C Seq L Contents
8
ID Type
Cntx P 0 L ("00000001")
1
Value
Cntx P Booln L Contents
3
Bind Token Signed Data
UNV P Bit Str L ("10011110")
1

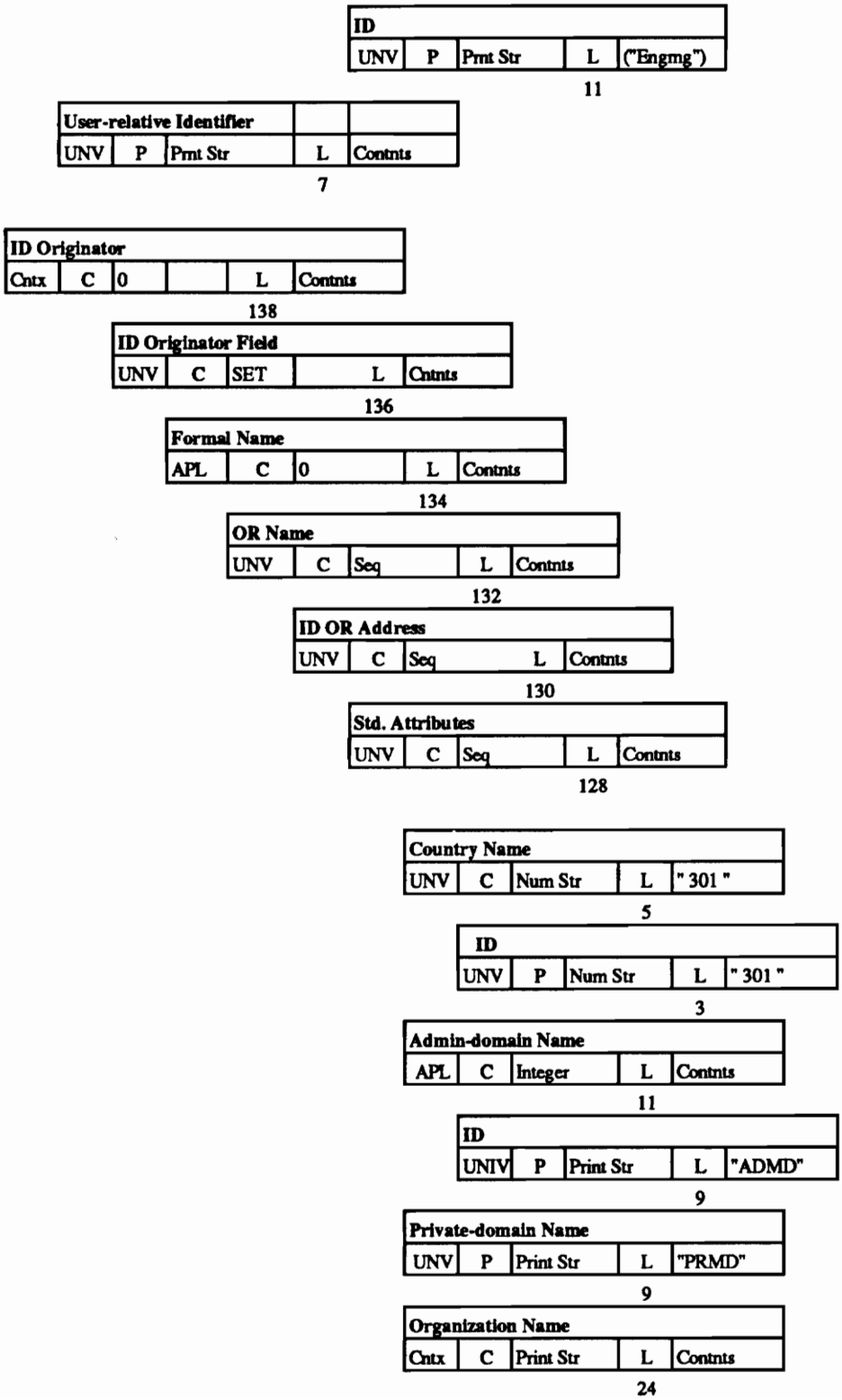
APPENDIX B

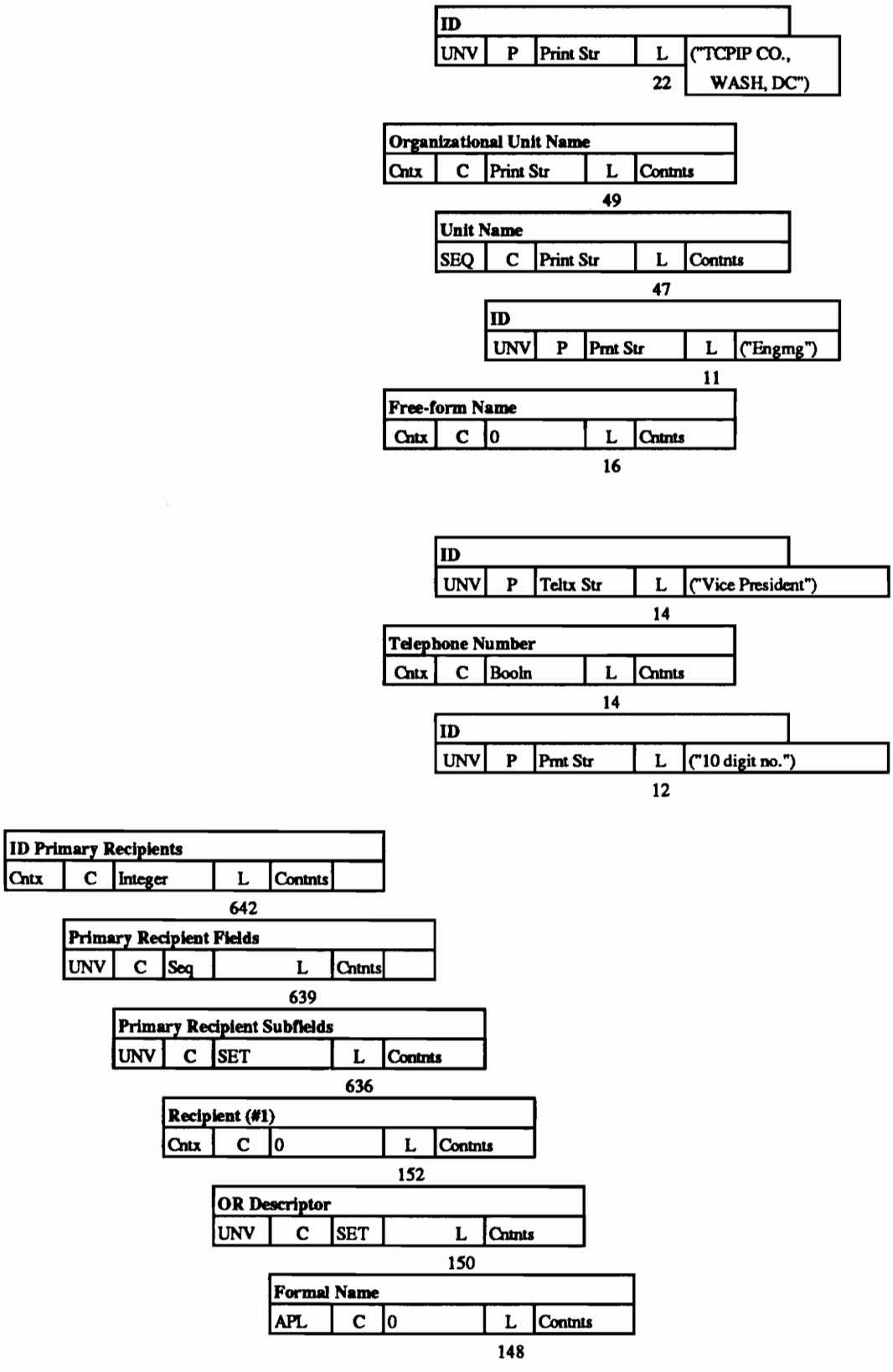
IPM HEADER ENCODING

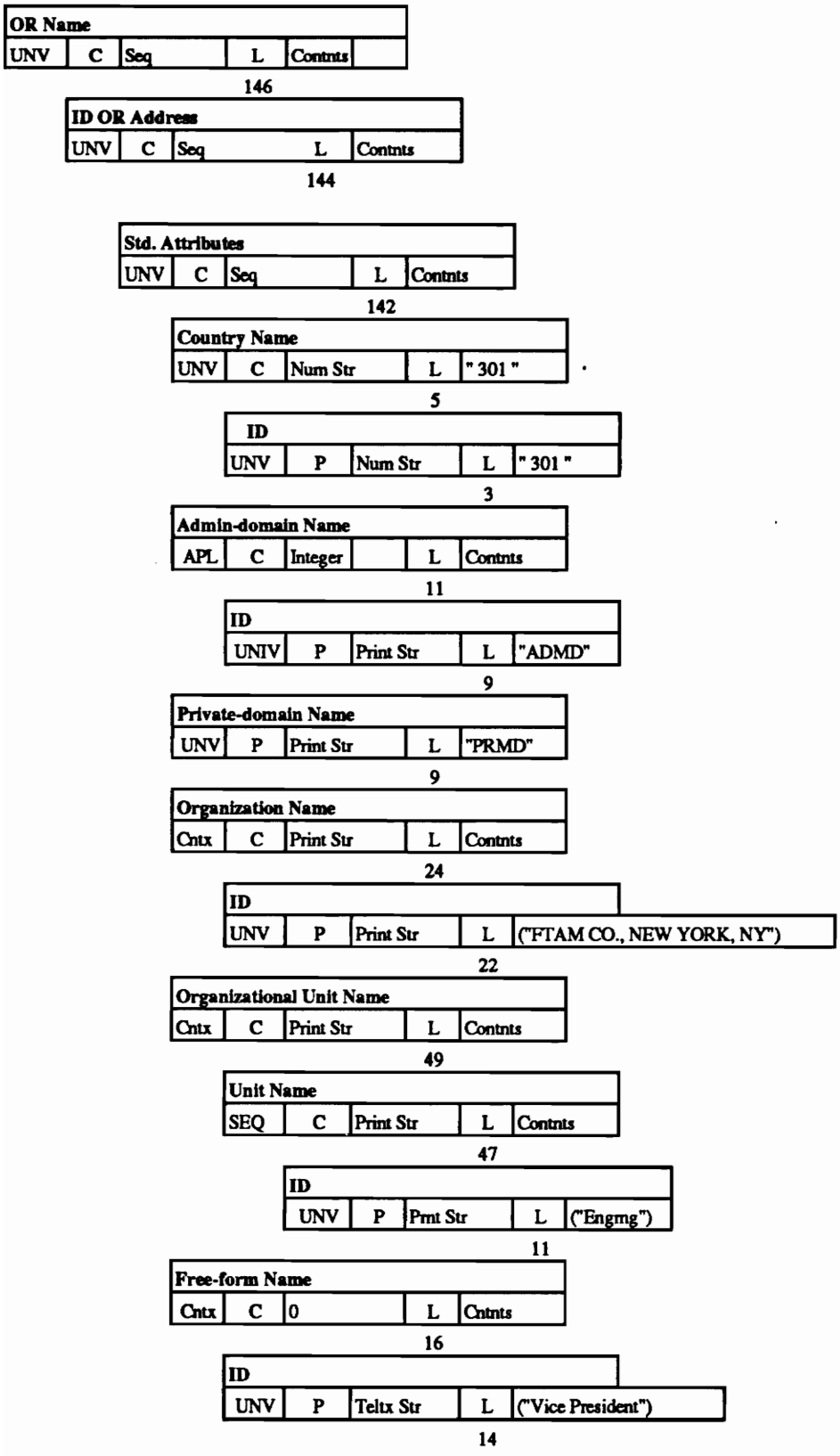
Total IPM Heading Length = this IPM + ID Originator +

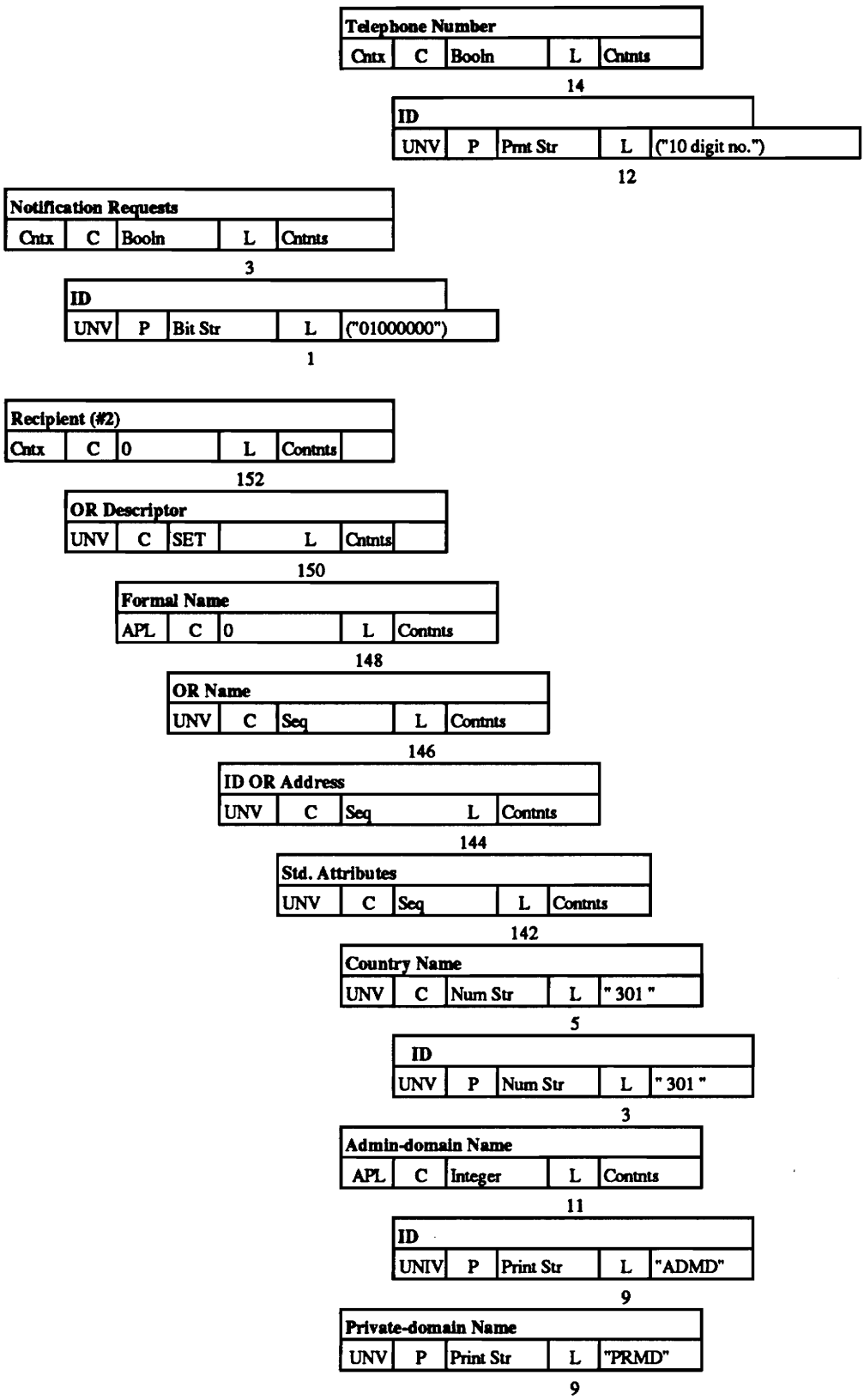
ID Primary Recipients = 2817 Octets

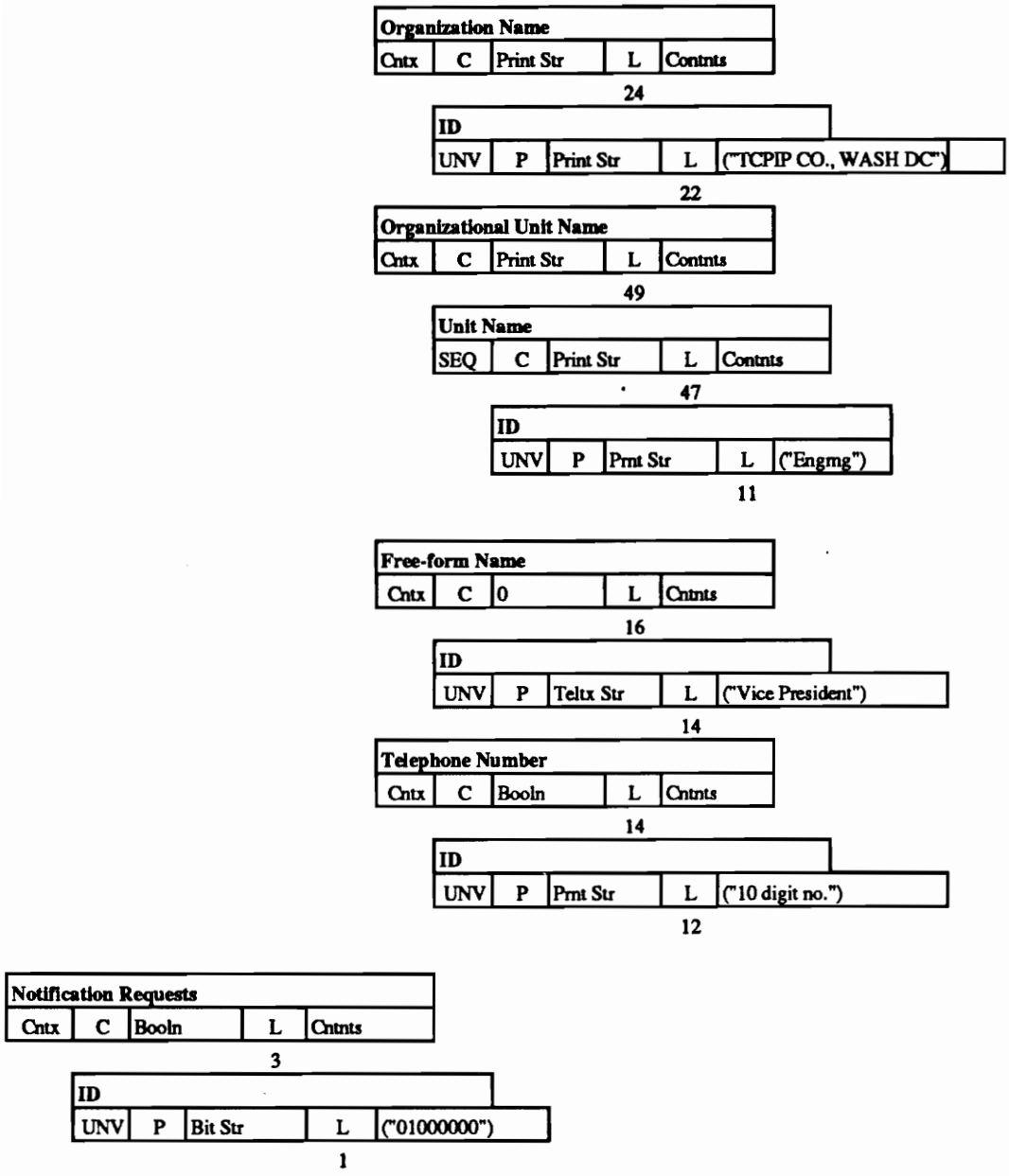



ID
UNV P Prnt Str L ("Engmg")
11
User-relative Identifier
UNV P Prnt Str L Contnts
7
ID Originator
Cntx C 0 L Contnts
138
ID Originator Field
UNV C SET L Cntnts
136
Formal Name
APL C 0 L Contnts
134
OR Name
UNV C Seq L Contnts
132
ID OR Address
UNV C Seq L Contnts
130
Std. Attributes
UNV C Seq L Contnts
128
Country Name
UNV C Num Str L " 301 "
5
ID
UNV P Num Str L " 301 "
3
Admin-domain Name
APL C Integer L Contnts
11
ID
UNIV P Print Str L "ADMD"
9
Private-domain Name
UNV P Print Str L "PRMD"
9
Organization Name
Cntx C Print Str L Contnts
24


ID
UNV
P
Print Str
L
("TCPIP CO., WASH, DC")
22
Organizational Unit Name
Cntx
C
Print Str
L
Contnts
49
Unit Name
SEQ
C
Print Str
L
Contnts
47
ID
UNV
P
Prnt Str
L
("Engrng")
11
Free-form Name
Cntx
C
0
L
Cntnts
16
ID
UNV
P
Teltx Str
L
("Vice President")
14
Telephone Number
Cntx
C
Booln
L
Cntnts
14
ID
UNV
P
Prnt Str
L
("10 digit no.")
12
ID Primary Recipients
Cntx
C
Integer
L
Contnts
642
Primary Recipient Fields
UNV
C
Seq
L
Cntnts
639
Primary Recipient Subfields
UNV
C
SET
L
Contnts
636
Recipient (#1)
Cntx
C
0
L
Contnts
152
OR Descriptor
UNV
C
SET
L
Cntnts
150
Formal Name
APL
C
0
L
Contnts
148


OR Name
UNV C Seq L Contnts
146
ID OR Address
UNV C Seq L Contnts
144
Std. Attributes
UNV C Seq L Contnts
142
Country Name
UNV C Num Str L " 301 "
5
ID
UNV P Num Str L " 301 "
3
Admin-domain Name
APL C Integer L Contnts
11
ID
UNIV P Print Str L "ADMD"
9
Private-domain Name
UNV P Print Str L "PRMD"
9
Organization Name
Cntx C Print Str L Contnts
24
ID
UNV P Print Str L ("FTAM CO., NEW YORK, NY")
22
Organizational Unit Name
Cntx C Print Str L Contnts
49
Unit Name
SEQ C Print Str L Contnts
47
ID
UNV P Prnt Str L ("Engrng")
11
Free-form Name
Cntx C 0 L Cntnts
16
ID
UNV P Teltx Str L ("Vice President")
14


Telephone Number
Cntx C Booln L Cntnts
14
ID
UNV P Prnt Str L ("10 digit no.")
12
Notification Requests
Cntx C Booln L Cntnts
3
ID
UNV P Bit Str L ("01000000")
1
Recipient (#2)
Cntx C 0 L Contnts
152
OR Descriptor
UNV C SET L Cntnts
150
Formal Name
APL C 0 L Contnts
148
OR Name
UNV C Seq L Contnts
146
ID OR Address
UNV C Seq L Contnts
144
Std. Attributes
UNV C Seq L Contnts
142
Country Name
UNV C Num Str L " 301 "
5
ID
UNV P Num Str L " 301 "
3
Admin-domain Name
APL C Integer L Contnts
11
ID
UNIV P Print Str L "ADMD"
9
Private-domain Name
UNV P Print Str L "PRMD"
9



So on for other recipients .....

References:

1. A Model for Computer Communications; U. Black, 1991, Prentice Hall, Engelwood Cliffs, N.J.

2. The Telecommunications Review; Summer 1991, MITRE Corporation, McLean, VA

3. OSI Explained; J. Henshall and S. Shaw, 2nd Edition, Ellis Horwood Ltd, NY

4. Data Communications and Interoperability; R. W. Markley, 1990, Prentice Hall, Englewood Cliffs, NJ

5. Local Area & Multiple Access Networks, Editor - R. L. Pickholtz, 1986, Computer Science Press, Rockville, MD

6. OSI Internet Performance; R. Wilder and A. Mankin, January 1991, MITRE Corporation, McLean, VA

7. ISO 8802-2 ANSI/IEEE Standard 8022.2; 1989, Information Processing Systems - Logical Area Networks - Part 2: Logical Link Control; ISO, Geneva, Switzerland

8. Evaluating the Viability of Fractional T1/T3; D. Palmer, November 1992, Networking Management

9. Beyond E-Mail: X.400 Messaging Applications; D. Traynor, Telecommunications, October 1992

10. X.25 Interface and End-to-End Virtual Circuit Service Characteristics; A. Rybczynski, IEEE Transactions on Communications, Volume COM-28, April 1980

11. Implementing the X.21 Interface; V. Yanoschak, February 1981, Data Communications

12. Internetwork Protocol Approaches; J. B. Portel, IEEE Transactions on Communications, Vol. Com-28, No.4, April 1980

13. Interfacing to a LAN: Where's the Protocol?; W. Stallings, Comp/Comm Consulting, Great Falls, VA

14. OSI Protocols Over SINCGARS Data Capability; G. A. Riechlen, September 1991, Working Paper MITRE Corporation, McLean, VA

15. OSI Application Protocols; A. Tang and S. Scoggins, Computer Science and Telecommunbications Program, University of Missouri, Kansas City

16 Analysis of X.400 Overhead; L. E. McArthur and K. C. Bryant, April 1991, Working Paper, Defense Information Systems Agency

17. Computer Networks, Second Edition; A. Tanenbaum, 1988, Prentice-Hall, Englewood Cliffs, NJ

Vita

The author, Sham Chakravorty, is currently employed as an engineer at MITRE Corporation in McLean, Virginia. He has over twenty years of experience in the field of telecommunications engineering and systems implementation. He holds BSCE and MSCE degrees and also completed an Electronics Engineering Diploma program. His professional affiliations include the The Institute of Electrical and Electronics, Inc. (IEEE). Mr. Chakravorty is forty two years old.

Sham Chakravorty