INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.630

(09/98)

SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

OSI networking and system aspects – Efficiency

Efficient open systems interconnection (OSI) operations

ITU-T Recommendation X.630

(Previously CCITT Recommendation)

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	**X.630–X.639**
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION X.630

EFFICIENT OPEN SYSTEMS INTERCONNECTION (OSI) OPERATIONS

Summary

This Recommendation provides the overview of the efficient protocol operations at the various layers identified by the reference model for open systems interconnection specified in Recommendation X.200.

Source

ITU-T Recommendation X.630 was prepared by ITU-T Study Group 7 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 25th of September 1998.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Recommendation X.630

EFFICIENT OPEN SYSTEMS INTERCONNECTION (OSI) OPERATIONS

(Geneva, 1998)

1 Scope

This Recommendation provides an overview of efficient protocol operations at the various layers identified by the Reference Model for Open Systems Interconnection specified in ITU-T Rec. X.200 | ISO/IEC 7498-1. It provides an overview to the set of Recommendations and International Standards that have been produced to facilitate the interconnection of information processing systems in an open environment where efficiency of communications is paramount. Such efficiencies include:

a) reduction in the overhead needed to encode control information for use in bandwidth-limited environments (such as radio links) or processing-limited systems (such as switching systems);

b) reduction in the delay to set up the association between the communicating applications so that data transfer can begin expeditiously; and

c) reduction in the support of unneeded functionality in certain environments where the communications requirements of the applications are limited.

This Recommendation should be useful to designers of OSI applications and implementors of OSI protocols for a variety of purposes which include serving as a refinement of the understanding of OSI, and extending the usefulness and applicability of these efficient OSI protocols to other application domains (e.g., real-time, and/or low bandwidth and/or long-delay environments).

2 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

2.1 Identical Recommendations | International Standards

– ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic reference model: The basic model.*

– ITU-T Recommendation X.207 (1993) | ISO/IEC 9545:1994, *Information technology – Open Systems Interconnection – Application layer structure*.

– ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology –Open Systems Interconnection – Basic Reference Model: Conventions for the definition of OSI services*.

– ITU-T Recommendation X.212 (1995) | ISO/IEC 8886:1996, *Information technology – Open Systems Interconnection – Data link service definition*.

– ITU-T Recommendation X.213 (1995) | ISO/IEC 8348:1996, *Information technology – Open Systems Interconnection – Network service definition*.

– ITU-T Recommendation X.214 (1995) | ISO/IEC 8072:1996, *Information technology – Open Systems Interconnection – Transport service definition*.

– ITU-T Recommendation X.215 (1995) | ISO/IEC 8326:1996, *Information technology – Open Systems Interconnection – Session service definition*.

– ITU-T Recommendation X.215 (1995)/Amd.1 (1997) | ISO/IEC 8326:1996/Amd.1:1998, *Information technology – Open Systems Interconnection – Session service definition – Amendment 1: Efficiency enhancements* .

– ITU-T Recommendation X.215 (1995)/Amd.2 (1997) | ISO/IEC 8326:1996/Amd.2:1998, *Information technology – Open Systems Interconnection – Session service definition – Amendment 2: Nested connections functional unit* .

– ITU-T Recommendation X.216 (1994) | ISO/IEC 8822:1994, *Information technology – Open Systems Interconnection – Presentation service definition*.

– ITU-T Recommendation X.216 (1994)/Amd.1 (1997) | ISO/IEC 8822:1994/Amd.1:1998, *Information technology – Open Systems Interconnection – Presentation service definition – Amendment 1: Efficiency enhancements* .

– ITU-T Recommendation X.216 (1994)/Amd.2 (1997) | ISO/IEC 8822:1994/Amd.2:1998, *Information technology – Open Systems Interconnection – Presentation service definition – Amendment 2: Nested connections functional unit.*

– ITU-T Recommendation X.217 (1995) | ISO/IEC 8649:1996, *Information technology – Open Systems Interconnection – Service definition for the association control service element*.

– ITU-T Recommendation X.217 (1995)/Amd.2 (1997) | ISO/IEC 8649:1996/Amd.2:1998, *Information technology – Open Systems Interconnection – Service definition for the association control service element – Amendment 2: Fast-associate mechanism*.

– ITU-T Recommendation X.225 (1995) | ISO/IEC 8327-1:1996, *Information technology – Open Systems Interconnection – Connection-oriented session protocol: Protocol specification* .

– ITU-T Recommendation X.225 (1995)/Amd.1 (1997) | ISO/IEC 8327-1:1996/Amd.1:1997, *Information technology – Open Systems Interconnection – Connection-oriented session protocol: Protocol specification — Amendment 1: Efficiency enhancements*.

– ITU-T Recommendation X.225 (1995)/Amd.2 (1997) | ISO/IEC 8327-1:1996/Amd.2:1998, *Information technology – Open Systems Interconnection – Connection-oriented session protocol: Protocol specification – Amendment 2: Nested connections functional unit*.

– ITU-T Recommendation X.226 (1994) | ISO/IEC 8823-1:1994, *Information technology – Open Systems Interconnection – Connection-oriented presentation protocol: Protocol specification* .

– ITU-T Recommendation X.226 (1994)/Amd.1 (1997) | ISO/IEC 8823-1:1994/Amd.1:1997, *Information technology – Open Systems Interconnection – Connection-oriented presentation protocol: Protocol specification – Amendment 1: Efficiency enhancements*.

– ITU-T Recommendation X.226 (1994)/Amd.2 (1997) | ISO/IEC 8823-1:1994/Amd.2:1998, *Information technology – Open Systems Interconnection – Connection-oriented presentation protocol: Protocol specification – Amendment 2: Nested connections functional unit*.

– ITU-T Recommendation X.227 (1995) | ISO/IEC 8650-1:1996, *Information technology – Open Systems Interconnection – Connection-oriented protocol for the association control service element: Protocol specification* .

– ITU-T Recommendation X.227 (1995)/Amd.2 (1997) | ISO/IEC 8650-1:1996/Amd.2:1998, *Information technology – Open Systems Interconnection – Connection-oriented protocol for the association control service element: Protocol specification – Amendment 2: Fast-associate mechanisms* .

– ITU-T Recommendation X.233 (1997) | ISO/IEC 8473-1:1998, *Information technology – Protocol for providing the connectionless-mode network service: Protocol specification*.

– ITU-T Recommendation X.234 (1994) | ISO/IEC 8602:1995, *Information technology – Protocol for providing the OSI connectionless-mode transport service*.

– ITU-T Recommendation X.235 (1995) | ISO/IEC 9548-1:1996, *Information technology – Open Systems Interconnection – Connectionless Session protocol: Protocol specification* .

– ITU-T Recommendation X.236 (1995) | ISO/IEC 9576-1:1995, *Information technology – Open Systems Interconnection – Connectionless presentation protocol: Protocol specification* .

– ITU-T Recommendation X.237 (1995) | ISO/IEC 10035-1:1995, *Information technology – Open Systems Interconnection – Connectionless protocol for the association control service element: Protocol specification*.

– ITU-T Recommendation X.633 (1996) | ISO/IEC 14700:1996, *Information technology – Open Systems Interconnection – Network fast byte protocol*.

– ITU-T Recommendation X.634 (1996) | ISO/IEC 14699:1997, *Information technology – Open Systems Interconnection – Transport fast byte protocol*.

– ITU-T Recommendation X.650 (1996) | ISO/IEC 7498-3:1997, *Information technology – Open Systems Interconnection – Basic reference model: Naming and addressing*.

– ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

– ITU-T Recommendation X.681 (1997) | ISO/IEC 8824-2:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.

– ITU-T Recommendation X.682 (1997) | ISO/IEC 8824-3:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.

– ITU-T Recommendation X.683 (1997) | ISO/IEC 8824-4:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*.

– ITU-T Recommendation X.690 (1997) | ISO/IEC 8825-1:1998, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*

– ITU-T Recommendation X.691 (1997) | ISO/IEC 8825-2:1998, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

– ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation*.

– ITU-T Recommendation X.881 (1994) | ISO/IEC 13712-2:1995, *Information technology – Remote Operations: OSI realizations – Remote Operations Service Element (ROSE) service definition*.

– ITU-T Recommendation X.882 (1994) | ISO/IEC 13712-3:1995, *Information technology – Remote Operations: OSI realizations – Remote Operations Service Element (ROSE) protocol specification*.

2.2 Paired Recommendations | International Standards equivalent in technical content

– ITU-T Recommendation X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements*.

ISO/IEC 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation Conformance Statements*.

– ITU-T Recommendation X.637 (1996), *Basic connection-oriented common upper layer requirements*.

ISO/IEC ISP 11188-1:1995, *Information technology – International Standardized Profile – Common upper layer requirements – Part 1: Basic connection oriented requirements*.

– ITU-T Recommendation X.638 (1996), *Minimal OSI facilities to support basic communications applications*.

ISO/IEC ISP 11188-3:1996, *Information technology – International Standardized Profile – Common upper layer requirements – Part 3: Minimal OSI upper layer facilities*.

– ITU-T Recommendation X.639 (1996), *Basic connection-oriented requirements for ROSE-based profiles*.

ISO/IEC ISP 11188-2:1996, *Information technology – International Standardized Profile – Common upper layer requirements – Part 2: Basic connection oriented requirements for ROSE-based profiles*.

2.3 Additional references

– Internet Engineering Task Force, Request for Comments 1698, *Octet sequence for upper-layer OSI to support basic communications applications.*

– ISO/IEC TR 10000-1:1998, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: General principles and documentation framework*.

– ISO/IEC TR 10000-2:1998, *Information technology – Framework and taxonomy of international standardized profiles – Part 2: Principles and taxonomy for OSI Profiles*.

3 Definitions

This Recommendation makes use of the following definitions.

3.1 Reference model definitions

3.1.1 Basic Reference Model definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.200 | ISO/IEC 7498-1:

a) application-entity;

b) application layer;

c) application-process;

d) application-protocol-control-information;

e) application-protocol-data-unit;

f) application-service-element;

g) application-association;

h) presentation-connection;

i) presentation layer;

j) presentation-service;

k) session-connection;

l) session layer;

m) session-protocol;

n) session-service;

o) transport layer

3.1.2 Naming and addressing definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.650 | ISO/IEC 7498-3:

a) application-process title;

b) application-entity qualifier;

c) application-entity title;

d) application-process invocation-identifier;

e) application-entity invocation-identifier;

f) presentation address;

g) presentation selector; and

h) session selector.

3.2 Service conventions definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.210 | ISO/IEC 10731:

a) primitive;

b) request (primitive);

c) indication (primitive);

d) response (primitive); and

e) confirm (primitive).

3.3 Presentation definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.216 | ISO/IEC 8822 and ITU-T Rec. X.226 | ISO/IEC 8823-1:

a) abstract syntax;

b) abstract syntax name;

c) default context;

d) defined context set;

e) [presentation] functional unit;

f) normal mode [Presentation];

g) presentation context;

h) presentation data value;

i) presentation selector;

j) transfer syntax; and

k) transfer syntax name.

This Recommendation makes use of the following terms defined in ITU-T Rec. X.216/Amd.1 | ISO/IEC 8822/Amd.1 and ITU-T Rec. X.226/Amd.1 | ISO/IEC 8823-1/Amd. 1:

l) null-encoding protocol option;

m) nominated context protocol option;

n) short-encoding protocol option;

o) packed encoding rules protocol option; and

p) short-connect protocol option.

3.4 Session definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.215 | ISO/IEC 8326 and ITU-T Rec. X.225 | ISO/IEC 8327-1:

a) session selector;

b) [session] functional units; and

c) orderly release.

This Recommendation makes use of the following terms defined in ITU-T Rec. X.215/Amd.1 | ISO/IEC 8326/Amd.1 and ITU-T Rec. X.225/Amd.1 | ISO/IEC 8327-1/Amd.1:

d) null-encoding protocol option;

e) short-encoding protocol option;

f) short-connect protocol option; and

g) no-orderly-release functional unit.

3.5 Application Layer Structure definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.207 | ISO/IEC 9545:

a) application-association;

b) application-context;

c) application-entity invocation;

d) control function; and

e) application-service object.

3.6 ACSE service definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.217 | ISO/IEC 8649:

a) application-association; association;

b) Association Control Service Element;

c) requestor;

d) acceptor;

e) association-initiator; and

f) association-responder.

This Recommendation makes use of the following terms defined in ITU-T Rec. X.217/Amd.2 | ISO/IEC 8649/Amd.2:

g) fast associate mechanism.

3.7 Taxonomy of profile definitions

This Recommendation makes use of the following terms defined in ISO/IEC TR 10000-1:

a) A-Profile;

b) profile requirements list;

c) profile implementation conformance statement; and

d) PICS Proforma.

3.8 Minimal OSI facilities definitions

This Recommendation makes use of the following terms defined in ITU-T Rec. X.638 | ISO/IEC ISP 11188-3:

a) application programmatic interface;

b) basic communications application;

c) mOSI stack;

d) stack; stack implementation; and

e) transport-provider

4 Abbreviations

This Recommendation uses the following abbreviations:

ACSE Association Control Service Element

APDU Application Protocol Data Unit

API Application Program Interface

ASN.1 Abstract Syntax Notation One

BCA Basic Communications Application

BER Basic Encoding Rules

CMIP Common Management Information Protocol

CULR Common Upper Layers Requirement

HDLC High-level Data Link Control

ICS Implementation Conformance Statement

ISDN Integrated Services Digital Network

ISP International Standardized Profile

LAPB Link Access Protocol B

mOSI minimal OSI upper layer facilities

MS Message Store

MTA Message Transfer Agent

OSI Open Systems Interconnection

PCI Protocol Control Information

PDU Protocol Data Unit

PDV Presentation Data Value

PER Packed Encoding Rules

PICS Protocol Implementation Conformance Statement

PPDU Presentation Protocol Data Unit

PRL Profile Requirements List

QoS Quality of Service

ROSE Remote Operations Service Element

RTSE Reliable Transfer Service Element

SI SPDU Identifier

SPDU Session Protocol Data Unit

TMN Telecommunications Management Network

TSDU Transport Service Data Unit

UA User Agent

VTP Virtual Terminal Protocol

5 Efficient OSI protocol operations

5.1 Background/motivation for studies of efficient OSI protocol operations

While many communications protocols have chosen to base their protocol architectures on the 7-layer OSI Reference Model, in many cases the actual protocol stacks have been "short", i.e. with missing layers. For instance, the ISDN network signalling standard, Signalling System No. 7, defined in the Q.700-series of ITU-T Recommendations, places the application layer directly on top of a specially engineered, highly reliable connectionless network service. The reason is that most signalling applications use a request/reply, single round trip communications and the overhead of upper layer connection establishment (indeed of the OSI upper layer services themselves) is seen as onerous.

NOTE – The term "upper layers" refers to the combination of the session and presentation layers, and ACSE which are used by all OSI-conformant application protocols.

Another example of "short stacks" is the Group 4 facsimile application which is defined in Recommendation T.90, and consists of X.75 layer 2 (similar to LAPB), the X.25 Packet Layer Protocol and the OSI transport and session layers but no support of ACSE and the presentation layer. The rationale in this instance is partly timing. The complete specification of all the OSI upper layer protocols were not available at the time Recommendation T.90 was approved; so the definition of the ACSE and presentation functions were folded into the Group 4 facsimile application protocol.

This example illustrates another problem, which is potentially of great concern to many applications. It pertains to the situation illustrated best by the use of Group 4 facsimile transmissions (which uses OSI protocols up through the session layer) over satellites. Data communications over satellite links involves unavoidable transmission delays that can impact the transmission efficiency of networks and the data processing efficiency at endpoints. A measure of the efficiency of a data communications protocol in this regard, therefore, is the number of round trip transmissions required to transmit a given set of data. The connection establishment procedures employed by the existing OSI protocols take too long (i.e. too many round trips prior to the start of data transfer) when operating over a one-hop or two-hop satellite circuit. The OSI upper layer connections cannot be established until the transport connection is set up end-to-end, which, in turn, requires the set-up of the underlying network connections. Thus, two or three round trips are required to complete the establishment of the association before data transfer may begin.

Therefore, such applications that might otherwise employ the full seven layer OSI protocol stack find that they do not need any of the additional services provided by some or all of the OSI upper layers, and that use of the current minimum conforming OSI protocols at those layers would be too inefficient for their current needs. At the time of their specification, having no efficient alternative, they "skipped" such layers as they deemed unnecessary and operated over "short stacks". This makes them non-compliant with OSI and limits their ability to easily adopt the use of the full OSI stack for communicating with other OSI-based applications that may (perhaps in the future) also need to be served using the same underlying transport mechanisms.

A common theme that has emerged when surveying the "short stacks" applications – and this is probably true of protocol stacks used in other industry sectors where communications is limited to a closed (albeit very large) group of users who communicate for specific purposes (often in real-time) – is that the OSI upper layers are seen as a "tool kit" from which piece parts are borrowed, not as something to be adopted as a whole. As examples of such adoption, the ASN.1 has been used to specify application layer protocols even though the only (until very recently) standardized encoding scheme – the Basic Encoding Rules (BER) – has been found to be very inefficient for real-time applications. The Packed Encoding Rules (PER), which provide highly compressed encodings while providing considerable encoding/decoding processing efficiency, is an alternative whose use appears to be more suitable for such environments. Similarly the OSI application layer protocol, the Remote Operations Service Element (ROSE), is used in ISDN signalling for defining switch-to-database queries/responses over a highly reliable network service for a large number of the so-called Intelligent Network services. However, in areas where OSI is pre-eminently suited – bulk-data transfer such as the transfer of files, for instance – the ITU-T Telecommunications Management Network (TMN) has adopted various OSI profiles for the support of the transfer of management information (like billing data, trouble reports, etc.).

To its advantage, the OSI Reference Model provides a complete upper layer architecture as opposed to the approach used by some applications, where the applications are placed directly at various transport addresses, which serve to identify their functions. The OSI upper layers, in a sense, provide a "tool box" of facilities which can be used by *any* application without the need to have every application design the same function in its own way. OSI's "mistake" might have been to design an upper layer architecture that is too feature-rich, whereas the needs of some of the simpler OSI applications (e.g. OSI Management) and other applications are considerably simpler.

Also, the manner in which the upper layer facilities have been collected into sets of functions makes it difficult to skip some functions entirely and to pick and choose the ones that an application really wants. OSI does permit a limited amount of picking and choosing through the use of functional units, but it is still not possible to skip some functions (e.g. Session kernel) entirely. In retrospect, therefore, one possibility for defining an "improved" upper layer architecture might have been to make every function optional. However, it must be recognized by designers of applications that are located directly on top of the transport layer that such applications may eventually need some, most, or all of these

functions. They will then be forced to decide if these should be done once to serve all (including future, as yet undefined) applications, or done individually for each application. If the former, then they will need to define an upper layer architecture that uses some or most of the modular functionality as provided by the current OSI upper layer protocols.

Given this environment, this Recommendation identifies various requirements (see 6.3) in the area of OSI upper layer protocol efficiency which, as addressed by the set of Recommendations and International Standards on efficient OSI protocol operations, has the potential of making the use of OSI upper layer protocols attractive to a larger number of applications.

5.2 Requirements and principles governing efficient OSI protocol operations

The following requirements have been identified for applications seeking efficient protocol operation while conforming to open systems standards. These requirements, depending on the scope of operation, are met by the protocol solutions described in clause 6 to address upper layer efficiency concerns:

1) **Improve efficiency of connection establishment:** Each end-to-end connection-mode OSI protocol currently requires a full round trip transmission for establishment negotiations prior to entering the data transfer phase. While the upper three layers' establishment are combined into a single round trip, a total of one to two of them are required by the Transport layer (including its use of the Network Services). Thus a total of two to three round trip transmissions are necessary before transfer of an application's data may commence. This greatly affects the efficiency of OSI communications, particularly over satellite connections (e.g. Group 4 fax over several satellite hops). Of lesser concern is the delay when connection establishment is refused or when the connection has to be released.

2) **Reduce PCI in all phases:** Many ITU-T applications operate in networks (such as radio links) where the bandwidth is limited, or with real-time processing-limited systems (e.g. switching systems) where long messages and less efficient encoding/decoding (e.g. the use of BER) cause a serious performance bottleneck. However, the reduction in PCI must be considered with the following points in mind: Is the PCI reduced:

   - through the use of more efficient encodings; and/or
   - tighter specifications (e.g. more succinct abstract syntax specifications); and/or
   - removal of redundant information or that which can be inferred by other means (e.g. prior knowledge); versus
   - those that come from a restriction or reduction in the layer services?

   NOTE 1 – Reduced PCI is unavoidably tied to the issue of interworking. If the bandwidth restrictions of the environment where the application operates are such as to require extremely short messages in the very first message of the establishment phase, then clearly the existing ACSE, presentation and session connection establishment PCI of 110 octets, where some of the octets contain fields permitting negotiation of versions, protocol options and optional functions, are unacceptable. Thus, full interworking through negotiation and extreme bandwidth efficiency in the first message are both not possible.

3) **Reduce unneeded functionality:** Some applications do not need all the functions that have been defined in the OSI upper layers to support the most general case. Even the existing minimum conforming OSI protocols at these layers are perceived to be too inefficient for the needs of these applications. Indeed, in many cases, even the so-called kernel functions are seen as an unnecessary overhead (e.g. for applications which do not require orderly release, have a well-defined single abstract/transfer syntax, are located at known addresses etc.). At this time, having no efficient alternative, they have taken to "skipping" such layers and operating over "short stacks". This makes them non-compliant with OSI and limits their ability to easily adopt the use of the full OSI stack for other OSI-based applications that may (perhaps in the future) also need to be served using the same underlying transport mechanisms. Thus, the concern here is to support the cases where such applications do not need these functions at all and are not simply reproducing the discarded OSI session and presentation layer functions as a part of the application protocol.

The following principles have also guided the development of the protocol solutions described in this Recommendation for improving OSI upper layer protocol efficiency:

1) **The OSI Reference Model must be supported:** That is, some PCI for the session layer, presentation layer and ACSE must flow during the application-association and session/presentation-connection *establishment* phase. This principle must not be understood as requiring consistency with the OSI Reference Model for its own sake; rather, the OSI Reference Model is seen as the protocol architecture which presents both an evolutionary target for migrant architectures as well as a single, uniform framework for describing the communications needs of varying sets of applications. The role of the OSI Reference Model in guiding, albeit in a piecemeal fashion, the protocol architectures developed by other applications must not be undermined.

   NOTE 2 – The interpretation of PCI must not be construed as identifiable "bits on the line." How the peer layer-entities communicate (i.e. whether by their own PCI or by the parameters of a supporting layers services) is not material, and both are compatible with the support of the Reference Model.

2) **Any efficient upper layer protocols defined must be extensible:** The new, efficient protocols should contain procedures and PCI to ensure interworking with any future extensions.

3) **The standardization and subsequent implementation effort to provide OSI upper layer protocol efficiency must not be too complex or lengthy:** Any solution proposed has the greatest chance of acceptance in terms of standardization and implementation if it provides an easy transition from existing standards (e.g. as an additional feature, a new protocol option, etc.) rather than introduce new concepts or radically different modes of operation. It must be noted, at the same time, that the "time to market" will be a critical factor in the acceptance of efficient OSI-based protocol operations instead of the development of diverging solutions by different user communities.

5.3 Interworking requirements for efficient OSI protocol operations

Another requirement guiding the development of efficient protocol operations for the OSI upper layers is the need to ensure interworking (to the extent possible) with existing OSI standards.

Indeed, a corollary to the requirement to support reduced functionality (see 5.2) is one that permits an open system employing protocols that support such reduced functionality (in support of a given set of applications) to communicate with peer applications/systems in an efficient manner for the majority of instances of communications. However, it would continue to employ the OSI upper layer protocols (as currently standardized) for certain other instances of communications in support of existing OSI-based applications. The point to remember is that interworking is in support of communications between like applications, and the specific cases described in this Recommendation deal with the situation where such applications could have been supported by the current OSI upper layer standards but find it preferable, for efficiency reasons, to use the protocols offering reduced functionality.

To this end, using the term "full" to denote the existing session/presentation/ACSE protocols, and "efficient" to denote any protocol that provides reduced or more efficient session/presentation/ACSE facilities, the following must hold (to the extent possible, given that the behavior of the existing OSI protocols are already defined):

NOTE 1 – The "full" protocols based on the existing Session/Presentation/ACSE specifications include the (fully-interworking) use of the mOSI functionality defined in ITU-T Recommendation X.638) – An open system may support both the "efficient" and the "full" protocols, i.e., be bi-capable. As an association-initiator, such a system can initiate an association using either a full-type protocol or an efficient-type protocol. In the most general case, which may not be the most efficient in every dimension, it may offer both and use the one acceptable to the responder. As an association-responder, the system must be able to distinguish a full-type association request from an efficient-type association request.

NOTE 2 – The initiator might know a priori, whether to use the "full" or "efficient" type association establishment (this can be done by the directory function, however implemented, or by always trying the "efficient" type, then retrying with the "full"). In the case where it offers both capabilities, in the hope that the responder will choose the "efficient" option if supported; otherwise, it falls back to the "full" option the first message must necessarily be understandable to the responder, which means that it must have the same bandwidth disadvantages of the current protocols.

a) An open system may only support the full protocol. As an association-responder, an existing mechanism (part of the full facilities) must exist to allow some notification of failure if it receives an efficient-type association request.

NOTE 3 – The existing upper-layer protocol specifications do not always mandate a defined reply to protocol violations; therefore the failure to understand an efficient-type protocol will usually not be communicated. However, such a defined response exists in Recommendation X.637, which requires that the Session REFUSE SPDU be sent to indicate that a new Session version is not supported or that the received SPDU has an SI field that does not match any of the currently standardized values.

b) An open system may only support the efficient facilities. The design of the new protocols should be such as to ensure certain failure, i.e. the system supporting the "full" capabilities will not confuse the new protocols with the existing ones. As an association-responder, a mechanism must be included as part of the efficient facilities that allows some notification of failure if it receives a full-type association request.

6 Efficient upper layer protocol operations

6.1 Introduction

The choice of any underlying protocol stack must be derived from the communication requirements of the applications that it is expected to support. Therefore, as an example, if the application is such that it has no requirements for almost error-free data transmission and sequencing of information sent and received, it may be possible to choose a purely connectionless seven-layer protocol stack to support its communications with a peer. On the other hand, if the application has requirements for check-pointing and recovery (to ensure efficient transmission of long files, for instance), a connection-oriented stack with the support of some session services is required.

The set of Recommendations and International Standards that have been developed to support efficient OSI upper layer protocol operation each serve a distinct range of applications, as each solution to improving efficiency has distinct limitations in functionality when compared with the full set of existing OSI upper layer capabilities. Therefore, in subclause 6.2, the complete set of capabilities of the upper layers are catalogued. Subclause 6.3 points out some of the efficiency concerns of the existing upper layers. Then, in subclause 6.4, each efficient protocol option as provided in the set of Recommendations and International Standards is described in terms of the functions it offers, thereby making it manifest which applications can make use of it.

6.2 Functions provided by the OSI upper layers

The following is a list of the common OSI upper layer facilities that add value to the transport service. The list has been written in a simplified manner (i.e. not in terms of the various formal OSI upper layer definitions, which are provided in parentheses where essential) to emphasize the role of the upper layers functions, not how it has been specified in terms of specific OSI layer services. Some of these facilities, as indicated, must be available in all open systems, either explicitly or implicitly, while others are optional. The facilities listed in items a) through j) are basic facilities whereas those listed in items k) through q) are more complex, optional facilities. All of them are intended to allow applications to be built without having to "re-invent" these facilities for each application:

a) identification of the state of the upper layer connections and association;

b) addressing information (session and presentation selectors), if necessary, to provide the local route within an end system to reach the appropriate application entity;

c) application layer addressing information, if necessary, (in the form of application-entity/process titles) which are location-independent;

d) negotiation of which application layer message sets (abstract syntaxes) will be used;

e) negotiation of what the communication will be about and policies governing that instance of communications (application context);

f) negotiation of the "bits on the line" (transfer syntax) for all the application data (abstract syntaxes) to be exchanged during an instance of communications. This capability further allows:

   i) unambiguous identification (via the presentation context) of each piece of application data as belonging to a particular message set (abstract syntax), even if two messages from different sets have identical "bits on the line"; and

   ii) alternative encodings, if desired, to be negotiated and used for the same message set.

g) enveloping of application data (via self-delimiting encoding rules) so that the encoded values need not contain length information, and concatenating them (distinguishably) for transmission in a single lower layer message;

h) the carriage, if necessary, of security information for authentication of partners as a part of the association establishment process, and, using extensibility, the inclusion of security mechanisms;

i) services, if needed, like checkpointing and resynchronization of the state of a dialogue, division of an instance of communications into well-defined portions (activities) etc.; and

j) the ability to negotiate versions of protocols, use of optional layer services (functional units) and ignore unknown information (via extensibility rules).

To this list could be added some additional, advanced features such as:

k) two-phase commitment and recovery;

l) the ability to multiplex higher-level associations on a single application association/session-and-presentation connection;

m) conveyance of remote procedure calls and operations and their return;

n) file access and transfer;

o) directory access and retrieval;

p) message transfer; and

q) transaction processing.

NOTE – The facilities k) through q) are often also called OSI applications, even though they are really "tools" which permit real user applications to be built.

6.3 Efficiency concerns regarding current OSI upper layer protocol

The current OSI protocols, particularly those defined for the upper layers, i.e., the layers above the transport service boundary, have acquired a reputation of being too complex, too large, too slow and too expensive to implement and test. One reason for this view is that most OSI applications currently defined use only a very small fraction of the facilities of the *standardized* upper layer facilities that are provided in 6.2. However, many implementations of upper layer protocols provide *every* upper layer facility specified, leading to complex implementations, and conformance tests thereof, for facilities which are rarely, if ever, exercised. This, in turn, causes *some* OSI upper layer implementations to be slow in comparison with proprietary protocols defined for specific applications.

The most inefficiencies occur in implementations that rigorously implement at layer boundaries. This naive approach is based on the mistaken requirement that the OSI layers correspond to parts of an implementation. The most probable cause for this misunderstanding is that, while the OSI Reference Model identifies unique n-entities for ACSE, and the presentation and session layers, the standards for these, when in operation, are "welded" together and consequently the protocol-engines (i.e. n-entities) for ACSE, the presentation, and session layers are tightly bound. That is, an application-association (using ACSE) is always jointly established and released with its supporting presentation-connection; a presentation-connection, in turn, is always jointly established and released with its supporting session-connection. An ACSE APDU is always encapsulated within an equivalent presentation PPDU (except for A-RELEASE); a presentation PPDU is always encapsulated within an equivalent session SPDU. Therefore, for example, as the session services are never used on their own, there is no point in having a separate session interface. Thus, there is no need to develop three separate protocol engines using three sets of service primitives and three sets of state tables defined in three sets of two Recommendations | International Standards for each layer.

One effort towards efficient OSI protocol operation, the minimal OSI functionality described in Recommendations X.637, X.638 and X.639, is primarily to encourage the implementation integration of the upper layers – session, presentation, and ACSE – to improve throughput performance. It is an effort, *based on the existing full connection-oriented upper layer standards*, to show that OSI implementations need not be large and complex. It does so by restricting the range of functions that an implementation has to support: so, if a particular facility is not required to serve an application, its implementation can be simplified, with the ability to fully interwork with the *same* application supported by the more comprehensive upper layer implementation.

For application-association/connection establishment, the *minimal* octet size for the existing composite upper layer PDU (Session, Presentation and ACSE PDUs) exceeds 110 octets. For this minimal, composite upper layer PDU, the only semantics transferred are the application context name for the association and the presentation context (abstract and transfer syntax names) of user data to be sent. The minimal overhead for sending the data transfer phase is 17 octets, where the only semantic is the length of the user data. For long duration applications or applications that need to exchange optional information (e.g. AE titles), the association establishment (and release) bandwidth (i.e. the size of the composite upper layer PDUs) may be acceptable. However, for many applications that operate in low-bandwidth environments such as radio links, such bandwidth overheads are not acceptable. The use of BER as the sole standardized way to encode application PCI specified in ASN.1 has been identified as one cause of excessive message lengths. The use of PER should alleviate this concern.

In fact, current upper layer protocols require the exchange of a number of association/connection establishment parameters which are not always needed or may be defaulted, particularly in the case where the communicating partners have a priori knowledge of the unsent information. Ideally, in such a case, the desirable composite upper layer PDU size could be made of a few octets. However, such gains can only come at the price of not being able to interoperate with existing OSI applications and implementations.

At present, the upper layer connection establishment awaits the successful set-up of the transport connection. It is not possible to save one round trip transmission per establishment of an OSI application association as the transport service cannot provide sufficient user data transmission capability, during the transport connection establishment procedure, in both directions, so that all of the upper layer connection establishment information (including "advanced user initialization information") can be embedded (i.e. placed as user data) within the transport connection establishment PDUs. Such embedding is currently not allowed because, in the most general case of use in unreliable subnetworks, the QoS of the Transport service with respect to error control during connection establishment may not be maintained at the same high level as is available during the transport data transfer phase. The ability to minimize the number of round trip transmissions required to release a Transport connection is also of importance, though not considered as essential as for the establishment phase.

6.4 Efficient upper layer protocol options

The upper layer facilities listed in 6.2, while useful in the general case, are not all necessary or not all used in specific environments, i.e. when used in support of particular applications.

In describing the approaches to efficiency that are available to applications, this Recommendation excludes the case where none of the facilities of 6.2 are used, or are duplicated in the application protocol so that the application is put directly on top of the transport (or an equivalent) service. Such cases, while often encountered, do not allow for a wider interworking community where such upper layer facilities are available for use in specific instances of communication but are not used in every instance.

To support open protocols within the OSI framework, this Recommendation describes an approach which systematically increases functionality starting from the minimal requirements of applications to reach the most general case for providing all the facilities defined in 6.2 for the communications needs of distributed applications. In this way, application designers may tailor their communications stacks somewhere along the spectrum to fit the current and future communications requirements of their applications.

6.4.1 Connection-mode protocols

Subclause 6.4.1.1 describes the most efficient subset of the OSI upper layer facilities to support applications whose requirements of upper layer services are minimal. This permits an almost "null" implementation of the upper layer stack with the ability to partially interwork with the more fuller-functionality stacks described in the subsequent subclauses.

Subclause 6.4.1.2 describes a mechanism, called "fast associate", which allows a session connection and the presentation connection/application association that it carries to be established using a compressed form of the information that would otherwise be sent on a session connection establishment. The compressed form references an upper layer context identifier (ulctx-id), which defines the values for options and requirements and any fixed fields and carries its own parameters for varying fields.

Subclauses 6.4.1.3 through 6.4.1.6 describe alternative protocol options for the session and presentation layers which can be used if an application indicates through its (abstract) service primitive interface that it has no need for some of the session and/or presentation facilities. While the use of these protocol options are only partially able to interwork with the full-functionality stacks, the use of these options provides both bandwidth and connection establishment efficiency. It comes closest to providing a migration path for applications which have missing layers to move in a direction which permits them to use efficient communications protocols (as defined here) for the majority of their instances of communications in support of their specific applications while permitting the possibility of supporting OSI-based applications on other (perhaps less frequent) instances of communications.

Subclause 6.4.1.7 describes the minimal OSI upper layer (mOSI) protocol stack to support applications which are typically single-purpose and with limited communications requirements for which the general-purpose implementations provide too many unneeded facilities, but which have the ability to fully interwork with the latter in support of the defined applications. Thin OSI, also described here, is a particular implementation of mOSI.

Subclause 6.4.1.8 describes the efficiency possibilities for proceeding beyond minimal OSI, should efficiencies be sought for the general-purpose, multi-applications, fully interworking OSI protocols.

6.4.1.1 Null-encoding protocol option

Null functionality at a layer refers to the case where no functionality is required of a layer during the data transfer phase but where OSI compatibility and compliance are required. This is the case which is most clearly applicable to applications which use "short stacks", to permit much greater OSI compatibility while at the same time allowing efficient communications. While it is possible to use the normal OSI layer protocol to signal that null functionality shall be required in the data phase, in certain instances, it is also possible to use a different protocol which is considerably more efficient (in terms of byte efficiency and, possibly, connection-establishment efficiency) to perform the negotiation. The term "fast byte" has been employed, as a convenient mnemonic, to refer to the insertion of a single byte PCI at connection establishment to signal that no further PCI will flow for that instance of communication. The use of the fast byte at a layer therefore serves to provide a service mapping between the layer above it and the layer below it.

NOTE 1 – Early discussions on the fast-byte concept considered the possibility of using the byte – theoretically only a bit – to serve as a *placeholder* so that it was possible, at some point during the instance of communication, to use the normal layer protocol to re-negotiate the use of some layer functionality where none was required earlier. At least for the upper layers, such a dynamic re-negotiation of layer functionality is for further study.

Thus, if a transport layer fast byte is exchanged, the layer service remains the same, i.e. the transport fast byte is a different version of the transport protocol with a one-to-one mapping of the network services to the transport services. In other words, by using the transport fast byte, one gets a QoS which is only as good as that provided by the underlying network service. The lower layer fast bytes are particularly useful for cases where the applications are communicating over a single data link, as is the case with ISDN access signalling.

For the upper layers, the typical, normal OSI implementation requires a 13- to 20-octet overhead on a single presentation data value (pdv) using the presentation and session data transfer services. This overhead is necessary to identify the state of the communication (i.e. that it is the data transfer phase as opposed to, say, the release phase), and to identify the pdv as belonging to a particular presentation context. Clearly, a null PCI optimization for the data phase implies a reduction in the layer service available to the application. For instance, in this case where all the application data is carried directly as user-data of the transport service, there is no guarantee that an encoded application pdu will not

resemble a session spdu; therefore, null PCI for the session data transfer phase implies that it is not possible to distinguish session spdus from application PCI. Therefore, it is not possible to use the orderly release facility of the session layer, though, of course, the application protocol can be defined to perform this function. Similarly, null PCI for presentation data transfer implies that there can only be one presentation context for the application pdus, whose abstract transfer syntaxes are known a priori. Thus, reducing the upper layer functionality inherent in the null functionality data phase restricts the range of applications that can use this optimization.

This loss of functionality must be reflected to the user at the service interface. For the session and presentation layers, the layer services are bundled together in groups known as functional units. At this time, orderly release of the session connection is provided as a part of the mandatory kernel functional unit. The use of null encoding for the data phase requires that the users have negotiated the use of a new functional unit, the so-called no orderly release functional unit, which removes the orderly release from the kernel functional unit.

NOTE 2 – The orderly release capability would more logically be a functional unit separate from the kernel; the new "negative" functional unit provides compatibility with the current specifications that require the (non-negotiable) kernel to be indivisible.

To this end, Recommendation X.216/Amd.1 defines the pass-through access to the session service, in particular the (new) no orderly release functional unit. As the presentation layer uses the session layer services for release of the presentation connection, there is no reduction to the presentation services. Thus efficiency optimizations available at the presentation layer are new protocol options, i.e. alternative, efficient PCI and procedures.

Recommendation X.226/Amd.1 defines two protocol options at the presentation layer that greatly reduce the quantity of presentation PCI in cases where the presentation user's requirements for presentation functionality are limited. The null-encoding protocol option provides an alternative presentation protocol option for data transfer with zero PCI which can be negotiated at connection establishment only if one of the following conditions described below is true:

a) the presentation context definition list contains precisely one item in which the abstract syntax name is known to the responding presentation protocol machine by bilateral agreement; or

b) the presentation context definition list is empty and the default context is known by bilateral agreement; or

c) the presentation context definition list is empty and the abstract syntax of the default context is known to the responding presentation protocol machine by bilateral agreement and is specified in ASN.1.

NOTE 3 – It may be possible in the future to negotiate the null-encoding protocol option for efficient data transfer using the presentation protocol defined in Recommendation X.226. It is left for further study to define an alternative version of the presentation protocol encoded using PER which will permit byte-efficient presentation negotiation of the full set of presentation functionality.

In addition, it is possible to use another protocol option, the short-encoding option, which defines encodings for some presentation PPDUs which are considerably shorter than the current ones if *both* conditions d) and e) described below are true.

d) the calling and called presentation selectors are null; and

e) the presentation-requirements parameter in the P-CONNECT service includes the kernel functional unit only.

The short-encoding protocol option allows the negotiation of the encoding rule which shall be used as the transfer syntax of the application PCI belonging to the single presentation context (which may be the default context) from one of BER, the aligned and unaligned variants of PER or a "transparent" encoding which is understood by bilateral agreement.

Recommendation X.215/Amd.1 specifies the no orderly release functional unit, whose selection by the session user indicates that the user has no requirements for orderly release of the session connection. Thus, either the application protocol has been chosen to perform this function, or the application association (which is one-to-one with the underlying session connection) is released by disconnecting the transport connection or by an abortive release of the session connection. The selection of this functional unit by the initiating session user permits the initiating session

protocol machine to offer the use of the null-encoding protocol option on the established session connection. The responding session protocol machine can accept this option if the responding session user has selected only (and nothing other than) the kernel, full-duplex and no orderly release functional units for use on the connection.

Recommendation X.225/Amd.1 describes how the negotiation of the null-encoding protocol option can be done using the protocol options field of the session establishment SPDUs defined in Recommendation X.225. However, Recommendation X.225/Amd.1 also defines the possibility of using the short-encoding protocol option for the establishment SPDUs, which define a one byte PCI for these SPDUs which are distinct from the leading octet of the current SPDUs, which provides a byte-efficient negotiation of the null-encoding protocol option provided that there is no session layer addressing information required to be exchanged, i.e. the session selectors are null.

It is expected that the short-encoding protocol option will be used in conjunction with the transport connection set-up to achieve interworking with current implementations and, for the case where the responder also implements this protocol option, achieve an improvement in round trip efficiency by setting up the upper layer connections concurrently with the transport connection. This is achieved as follows: the SHORT CONNECT SPDU – which is the short-encoding version of the current session CONNECT SPDU – is sent as user data of the T-CONNECT request service primitive. This requires that the SHORT CONNECT SPDU plus any accompanying user data meet the 32-octet limitation on the size of the transport user data.

NOTE 4 – The transport protocol class 0 does not permit the carriage of user data. Therefore, for this scenario to work, the transport protocol class 4 should be available at both ends, or the transport fast byte protocol should be employed.

Current session implementations ignore any user data on the T-CONNECT indication primitive, or, at worst, disconnect the transport connection. Thus, absence of any user data on the T-CONNECT confirm primitive is a signal to the initiating session protocol machine that the responder is an implementation of the current standards. If the responding session entity implements the short-encoding protocol option, the SHORT ACCEPT SPDU is sent as user data of the T-CONNECT response service primitive, and its receipt by the initiating session protocol machine completes the session connection establishment in tandem with the transport connection establishment.

Of course, the short-encoding option may be used with the T-DATA service for the case where an already established transport connection is assigned to the session connection. Interworking is not fully achievable as there is no guarantee that the responding session entity, if based on the current standards, will send a REFUSE SPDU to signal a protocol error, which is what a short-encoding for an SPDU would be.

6.4.1.2 Fast-associate/upper layer context identifier mechanism

The fast-associate mechanism allows a session connection, including its embedded presentation connection and application association, to be established using a compressed form of the information that would otherwise be sent on the S-CONNECT exchange. The compressed form, called the upper layer context identifier, is a reference to an upper-layer context specification, which is a definition of the fields of the application, ACSE, presentation and session protocols that would be sent on the full-form connect messages. The upper layer context identifier may be parameterized to include values for the variable fields allowed by the full form protocols for the upper layers.

6.4.1.3 Short-encoding mechanism

6.4.1.3.1 Session layer

The short-encoding mechanism is an independent optimization that reduces the session pci in general. The most commonly used Session Protocol Data Units (SPDUs) are given alternative encodings that can be used when the user requirements permit. The SPDUs used by the fast associate mechanism (see 6.4.1.2) are among these, while the others are only available when their use has been negotiated during session connection establishment.

6.4.1.3.2 Presentation layer

This is a protocol option for the presentation protocol which defines an alternative header encoding for the user-data of most presentation primitives when this contains only presentation Data Value (PDV). When used in combination with the session short-encoding option (see 6.4.1.3.1), it minimizes the overhead for this common case, without loss of overall function, as the short encoding can be freely mixed with the non-compressed encodings.

6.4.1.4 Short-connect protocol option

The short-connect option conveys only a single octet to mark the connect PDU. In this protocol option, presentation context is known and agreed by both users. The conveyance and use of upper-layer selectors is not enabled. The exchange is limited to the selection of transfer syntax. The exchange may, in fact, convey user-data.

6.4.1.5 Presentation/PER protocol option

In this Presentation protocol option, the use of the Packed Encoding Rules (PER) to encode the presentation PPDUs allows the full range of values for the presentation protocol to be sent using a smaller overhead in terms of the bits to be sent.

6.4.1.6 Nominated context protocol option

The nominated context presentation protocol option allows the "simple-encoding" choice for the presentation user data to be used for one of the presentation contexts of the Defined Context Set (DCS) even when there are many elements in the DCS. This allows a significant reduction in the presentation pci overhead for any application that sends most of its data in the same presentation context without any loss of functionality.

6.4.1.7 Minimal OSI/thin OSI

6.4.1.7.1 Minimal OSI (mOSI)

There is no novel standardization done in Recommendations X.637, X.638 or X.639 in providing minimal functionality for the OSI upper layers. These three Recommendations deal with defining a particular, minimal conforming set of options and procedures – called mOSI – taken from the base ACSE, presentation and session service and protocol standards, so that the result is a fully-conforming specification which leads to implementations that are inter-operable with those based on the complete upper layer specifications (the base standards) so long as the communicating applications' requirements are met by the mOSI facilities. mOSI is defined to support only a certain range of applications, the *basic communications applications*. Such applications are those which only have the ability to open an association, optionally negotiate the data representation, send/receive data, and close or abort the association. By restricting the use of mOSI to the support of basic communications applications, the mOSI specifications are therefore not concerned with those parts of the base ACSE, session and presentation protocols specifications that, while defined in the standards, will not be exercised by the application. Thus, mOSI-based implementations achieve their compactness by only implementing the parts that their applications need. This is just the regular profile or application dependency requirements of an application layer standard.

mOSI also ensures simpler implementations by eliminating the accurate analysis of invalid received protocol. mOSI works on the assumption that useful open interworking only requires the *sending* of *valid* protocol and the interpretation of *any valid received* protocol. Thus it spares itself a lot of implementation of software that, in support of some currently-specified conformance tests, reacts suitably to the receipt of invalid protocol.

As mOSI is based on the existing upper layer standards, there are no gains in terms of reduced PCI and connection establishment efficiency. mOSI was not designed to address these dimensions. It was designed to provide a single specification of a set of upper layer facilities which could serve the needs of the overwhelming majority of OSI-based applications.

Recommendation X.638 provides an overview of the minimal OSI upper layer facilities and the range of applications that can be supported by it. mOSI consists of the kernel functional units of ACSE, presentation and session services plus the session full-duplex functional unit. It provides optional support for two ACSE functional units: the authentication

functional unit, and the application context negotiation functional units. This choice is presumed to include the majority of OSI-based applications, which are written specifically to use the above-mentioned OSI layer services. Some examples are:

a) applications which do not use the services of the RTSE, e.g. UA-MS transfer of X.400;

b) FTAM implementations which do not use recovery;

c) TP implementations which do not use two-phase commitment;

d) VTP without the destructive interrupt facility; and

e) all ROSE-based applications such as the Directory Access Protocol and CMIP.

The set of connection-oriented applications needing more than basic communications facilities, from an OSI perspective, includes those applications which use one or more of the following facilities: session major or minor synchronize; resynchronize; activity management. Some examples of such applications which cannot make use of the mOSI facilities are:

a) RTSE-based applications, e.g. MTA-MTA transfer of X.400;

b) FTAM with optional recovery;

c) TP with optional two-phase commitment (i.e. CCR); and

d) some optional aspects of VTP.

Recommendation X.638, which is technically equivalent to ISO/IEC ISP 11188-3, is mainly in the form of a normal profile, and largely comprises of the answers to the questions in the PICS of the ACSE, presentation and session protocols. It also provides some guidance on the specification of an API that permits the so-called migrant applications, i.e. those that have been defined for use with some other communications stack, to use the mOSI facilities.

Recommendation X.637, which is technically equivalent to ISO/IEC ISP 11188-1, restricts the range of flexibility allowed in ACSE, session and presentation standards. To ensure interworking, an implementation based on mOSI must be able to receive all valid alternative encodings that might be sent to it by an implementation based on the full set of capabilities defined in the base standards. Recommendation X.637 provides restrictions on the maximum sizes of some fields and reduces some of the encoding flexibility of BER, thereby permitting the implementation of a simpler decoder.

Recommendation X.639, which is technically equivalent to the ISO/IEC ISP 11188-2, describes the common upper layer requirements above and beyond those specified in Recommendation X.637 in support of ROSE-based profiles which do not use the services of the RTSE. These common elements are specified by reference to the OSI connection-mode Recommendations for the ROSE protocol, the ACSE protocol, the presentation and session layer protocols.

6.4.1.7.2 Thin OSI

The size and efficiency of an implementation of OSI (any OSI implementation, not just of mOSI) is significantly affected by the implementation design. The OSI 7-layer model is an abstract model to promote layered specifications but, in many cases, may not be the best way to implement OSI. The services and protocols for the upper three layers of OSI are specified separately. However, considerable efficiency can be gained if all three protocol machines are combined as one module and not as three separate modules requiring the definition of formal interfaces between them. The three upper layer protocols provide an integrated set of services; these services are not useful individually.

NOTE 1 – Currently there are no formally defined APIs for the session and presentation service interfaces.

Combining layer protocol machines applies not only to mOSI implementations, but also to all OSI upper layer implementations. However, combining layers excludes the possibility of testing them individually. Individual layer testing is not recommended.

mOSI supports uncomplicated application protocols, i.e. byte stream or simple record oriented data transfer which probably constitutes the majority of user application protocols. Therefore, the encoding of embedded PDU headers (i.e. PCI) for all three protocols is uniform. Using predefined protocol headers allows significant gains in processing efficiency. This is especially true for data PDUs – the PDUs that generally constitute the majority of PDUs exchanged.

An OSI upper layer protocol machine that only supports the facilities defined in mOSI need only recognize a very limited subset of all potential upper layer PDU sequences. In particular, they need not recognize any non-mOSI protocol sequences and these may be treated as unrecognized PDUs. The receipt of an unrecognized PDU results in a protocol error and in the release of the association. This does not lessen the utility of the implementation but it does reduce the amount of code needed for error handling.

NOTE 2 – It is always legal for an OSI protocol machine to abort an association.

The Internet RFC 1698 – often referred to as the ThinOsi Upper Layer Cookbook – is an example of explicit predefined PDU encodings for mOSI implementations. The RFC contains a full description of this technique including BER encodings for PDUs. It provides particular octet sequences that comprise the OSI ACSE, presentation and session protocol when used to support basic communications applications. An implementation that sends the octet sequence given here, and interprets the equivalent protocol received, will be able to interwork with an implementation based on the base standard, when both are being used to support the same basic communication application.

NOTE 3 – The OSI standards (especially Presentation) permit considerable flexibility in the encoding of PDUs using the BER. The ThinOSI Cookbook defines particular octet sequences to be sent and describes variations that can be expected in the data received from an implementation based directly on the OSI upper layer base standards. If the implementation is able to interpret these received variants, then it can be expected to interwork with an implementation based on the base standards.

6.4.1.8 Beyond minimal OSI

It must be made clear at the outset that the general purpose, full functionality OSI upper layer stack is clearly unsuited for environments where the 100+ octet message size of the first connection establishment message and the need for two (or three) round trips for connection establishment before data transfer may begin are seen as onerous. The message size is the result of having full openness, while the seemingly excessive handshaking ensures reliability of the subsequent data exchange across multiple subnetworks of varying, unknown QoS.

However, even in such a case, it is possible to take advantage of the negotiation facilities in the upper layer protocols to ensure greater efficiency of operation than might be supposed. Two such instances are described below.

6.4.1.8.1 Application context name negotiation

Until recently, the application context negotiation mechanism in ACSE permitted the association initiator to offer a single application context name which the responder could either accept or, if unacceptable, offer an alternative name. If the alternative name was unacceptable to the initiator, the association was released. Otherwise, the name proposed by the responder was the context in place during the lifetime of the established association. In the case of systems communicating frequently, this negotiation presumably was concluded satisfactorily in one exchange. However, in the first-time communications with a system of unknown capabilities, it was possible that the establishment of a common, acceptable application context could take several exchanges.

To reduce the likelihood of more than one exchange to complete the application context negotiation, the current specification of ACSE offers a new functional unit – the application context negotiation functional unit – which, if selected by the initiator and accepted by the responder, permits the initiator to offer a list of application context names that it can support for that instance of communication, with the responder restricted to choosing one from that list. This, presumably, provides statistical efficiency in ensuring successful association establishment.

6.4.1.8.2 Transfer syntax negotiation

The use of BER as (until recently) the only standardized encoding rule, and its prescription as the transfer syntax for almost all OSI application protocol obscures the fact that the presentation protocol permits the negotiation and subsequent transfer of different encodings of the same abstract syntax values (message set).

NOTE 1 – The Presentation protocol places no restrictions on the use of any notation other than ASN.1 to define the abstract values. Indeed, many applications may not formally distinguish their abstract and transfer syntaxes, but such a distinction is usually possible – at least theoretically.

The Presentation protocol permits an initiator to offer an alternative (set of) transfer syntax(es) for any abstract syntax. To ensure interworking, most OSI-based applications require that the BER always be available. However, in situations where there is some knowledge of the capabilities of the communicating entities, the initiator can also offer some proprietary transfer syntax – designed presumably to improve bandwidth and/or processing time – which is known only within a smaller community. If the responder understands the efficient transfer syntax, it accepts its use. Otherwise, the default transfer syntax, BER, is used.

NOTE 2 – The use of alternative encodings do not imply standardization of such rules. As the transfer syntax name is an OBJECT IDENTIFIER, the registration of such a syntax can be devolved to any organization.

The recently standardized PER offers a highly compressed encoding with, as an unintentional byproduct, a greater efficiency in encoding and decoding than was possible with BER. There are a few requirements to be met and changes to be made if PER is to be used as the efficient transfer syntax for current OSI-based application protocols. The reason is this: PER is able to achieve its remarkable compactness because it does not encode *any* redundant information. It is able to do so only because it assumes that the two communication application entities share the *exact same* abstract syntax definition. Thus, for example, it can send simply two bits to identify an element in a CHOICE type containing four alternatives, as opposed to the one octet tag that would be required by the BER. It can leave out length fields where the size of the value is known from the abstract syntax definition, such as, for example, a constrained INTEGER. A PER-based encoding carries a single bit as the preamble of an encoding to identify if there is any extension marker in the notation. (A BER-based decoder simply ignores or skips over any additional data values, if unknown to it.) Therefore, existing specifications which use the ASN.1 notation (if they have extensibility defined in accompanying text or comments) cannot be used with PER until extensibility markers are added to the specification.

In more detail, for an existing specification using ASN.1, which will have been written when the extensibility markers were not part of the ASN.1 notation, the changes required are as follows:

a) if no extensibility is mentioned in the text or in comments, and none is required, no change is needed, and the standard can be used unchanged with PER;

b) otherwise, because textual description is imprecise, different implementors might assume different positionings of the extensibility markers based on the text and would produce non-interworking PER implementations; the addition of extensibility markers must be undertaken by the "owner" of the specification (e.g. a Recommendation or Standard), not the implementor;

c) for most such specifications, the required extensibility can be expressed using the extensibility markers; this will require an "editorial" revision of the specification; the revision will be fully compatible with current BER implementations, and will not affect their operation or the bits on the line, but will allow the specification (and its future revisions) to be used with PER;

d) for some specifications, the necessary extensibility requirements go beyond those which the ASN.1 standard requires to be supported by all encoding rules; in this case, the current specification is unavoidably unsuitable for use with PER (and potentially unsuitable for use with anything other than BER).

   NOTE 3 – Such an extensibility requirement alluded to in item d) might be the ability to add new elements at any place within an existing SET or SEQUENCE.

6.4.2 Connectionless-mode upper layer protocols

Some applications are best served by connectionless *functions* and, in some cases, the *communications* are best served by the use of connectionless lower layer functions. This subclause will consider the efficient application of connectionless functionality in the upper layers. The OSI RM places constraints on the use of connectionless functions at the various layers. In particular, the OSI RM prohibits the mapping between the connectionless-mode and the connection-mode at any except the Transport/Network layer boundary and in the Application layer. Therefore, this subclause considers the use of the connectionless-mode in the upper layers (see 7.3.3 for its application to the lower layers).

The upper layer connectionless-mode protocols have been purposely restricted to have very little functionality. Each connectionless protocol for each of the upper three layers provides a protocol identifier, a protocol version and addressing selectors. Since the address selectors can be assigned to "null" in most cases, the connectionless protocol overhead can be made sufficiently small that it does not warrant the creation of special "efficient" versions of the connectionless protocols. The following subclauses consider each of the upper layers in turn (treating the ACSE as, for this purpose, a "layer").

6.4.2.1 Connectionless-mode Session protocol

In virtually all uses of the connectionless Session protocol, there should be no requirements for using Session selectors. They should be assigned to NULL.

NOTE – Application designers who believe that they have a use for session selectors should consult an expert in their use, when it is likely to be shown to be unnecessary.

6.4.2.2 Connectionless-mode Presentation protocol

In virtually all uses of the connectionless Presentation protocol, there should be no requirement for using Presentation selectors. They should be NULL.

In rare cases, it may be necessary to include abstract and transfer syntax identifiers which can decrease byte efficiency. However, in most cases where the connectionless mode is used, they can be avoided by means of ad hoc arrangement, i.e. agreements outside the scope of protocols in profiles or implementation agreements.

6.4.2.3 Connectionless ACSE protocol

The requirements for using the connectionless-mode ACSE will be heavily dependent on the application's needs. For many telemetry-like applications where, for all intents and purposes, address assignments and contexts are static, the application (association) context, application-entity-title and application-service-object-title may be omitted. This will reduce the connectionless ACSE APDU to simply the protocol identifier and the version information.

6.5 Matching efficiency needs to various upper layer protocol options

The selection of various upper layer protocol options is motivated by the efficiency needs of various applications. The applications may be characterized as being one of three types.

The first type is legacy OSI applications as defined in 6.4.1.7.1 (i.e. RTSE users, FTAM with user recovery, TP with two-phase commit, VTP with destructive interrupt). These applications cannot use the efficiency enhancements.

The second type is classic OSI applications seeking general efficiency of encoding. To this end, the fast associate/upper layer context identifier and the short-connect option may usefully be applied. This may be the case with newer implementations of TP.

The third type is OSI applications that use only the kernel and full-duplex session functional units, ROSE-based applications such as directory access protocol and CMIP. These protocols may be supported by the efficiency enhancements. The null-encoding and short-connect protocol options are the preferred combination for any application of this type. It is important to note that the upper layers still offer important functionality in this combination through the use of ACSE.

The generality of an application is inversely proportional to its expected use of the efficiency enhancements. An application designed to be fully interoperable must offer the full negotiation capability in the initial connect exchange (including the use of BER).

An application that is domain-specific will often know the presentation context by prior agreement.

The use of highly optimized data encoding (e.g PER-visible extensibility markers) may require greater processing in the end systems but then offer large gains in use of link capacity. The trade-off of processing versus link capacity utilization must also be considered.

The use of the connectionless upper layers may be useful for applications with static PCI.

7 Efficient lower layer protocol operations

7.1 Introduction

Efficiency concerns for the Network and Transport Layers have focused on the need to completely eliminate the round trip propagation delays associated with the establishment and release of these protocol layers, and the need to minimize PCI during all phases of operation of these layers.

At present, the Network Layer waits for the successful establishment of the Data Link Layer connection before initiating a connection establishment request at the Network Layer. This adds one more round trip before the completion of the network connection.

7.2 Functions provided by the OSI Lower Layers

In addressing efficiency considerations at the Lower Layers, it is assumed that efficiency of operation is of primary concern, and that *enhancements* to Quality of Service are not required.

7.2.1 Data Link Layer

Even in the efficient mode of operations, the Data Link layer will continue to provide the capability for error free transmission at this layer and ensure that the frames arrive in the order in which they were transmitted (i.e. the sequence is maintained).

7.2.2 Network Layer

In order to eliminate the round trip propagation delay associated with the establishment and release of a network connection, and require very low PCI overhead, the following functions of the Network Layer are therefore not supported:

a) multiplexing;

b) error detection;

c) error recovery;

d) flow control;

e) expedited data;

f) encryption;

g) accounting mechanisms;

h) status exchanges and monitoring of QoS;

i) temporary release of data link connections.

The OSI Connection-mode Network Service defined in ITU-T Rec. X.213 | ISO/IEC 8348 is supported with the following restrictions:

1) the optional receipt confirmation and optional expedited data services are not supported; and

2) no enhancement of the Data Link service QoS is provided, so that the Network service QoS approximates to the corresponding Data Link service QoS.

7.2.3 Transport Layer

In order to eliminate the round trip propagation delay associated with the establishment and release of a transport connection, and require very low PCI overhead, the following functions of the Transport Layer are therefore not supported:

a) multiplexing and demultiplexing;

b) error recovery;

c) concatenation and separation;

d) splitting and recombining;

e) encryption;

f) accounting mechanisms;

g) status exchanges and monitoring of QoS;

h) blocking;

i) temporary release of network connections;

j) checksum.

The OSI Connection-mode Transport Service defined in ITU-T Rec. X.214 | ISO/IEC 8072 is supported with the following restrictions:

1) the length of TSAP IDs is fixed at 2 octets;

2) no enhancement of the Network service QoS is provided, so that the Transport service QoS approximates to the corresponding Network service QoS.

7.3 Lower layer protocol options

7.3.1 Data Link Efficiency Option

Enhancements have been made to the HDLC suite of protocols (which are used as the basis of many Data Link layer protocols) to allow for the inclusion of an information field in the mode setting/clearing commands. The Data Link layer service definition, Recommendation X.212, has been enhanced to allow for the inclusion of an information field during the link connection and release. The amount of information that can be included in the mode setting/clearing commands is limited to the maximum frame size that can be transmitted on the data link.

Therefore, with this enhancement it is possible to include the Network Layer Connect Request in the information field during the link establishment phase and similarly the Network Layer Disconnect can occur during the link release.

7.3.2 Fast byte for the Network and Transport layers

Two protocols, the Network Fast Byte Protocol and the Transport Fast Byte Protocol, have been developed in order to address efficiency considerations in the Lower Layers.

Unlike traditional protocols, the Network and Transport Fast Byte protocols do not define different PDU types (e.g.: connect, release, reset, etc.). Theses protocols define a single PDU, and the semantics of this PDU are dependent on the service primitive in which the PDU is received.

7.3.2.1 Network Fast Byte Protocol

The Network Fast Byte Protocol (Recommendation X.633) applies to the provision of the OSI connection-mode network service in end systems, eliminates the round trip delay associated with the establishment and release of a network connection, and requires very low PCI overhead. The Network Fast Byte Protocol is required for use in situations in which enhancements to the data link QoS are not required, and efficiency of operation (e.g. reduction of round trip delays on establishment and release) is of primary concern. The protocol ensures an interoperable method for accomplishing this, by standardizing a "mapping" between the network and data link services.

7.3.2.2 Transport Fast Byte Protocol

The Transport Fast Byte Protocol (Recommendation X.634) eliminates the round trip delay associated with the establishment and release of a transport connection, and requires very low PCI overhead. The Transport Fast Byte Protocol is intended for use in situations in which enhancements to the network QoS are not required, and efficiency of operation (e.g.: reduction of round trip delays on establishment and release) is of primary concern. The protocol ensures an interoperable method for accomplishing this, by standardizing a "mapping" between the transport and network services.

7.3.3 Efficient use of the connectionless mode

Some applications are best served by connectionless functions and, in some cases, the communications are best served by use of the connectionless lower layer functions. This subclause will consider the efficient application of connectionless functionality in the lower layers. As noted in 6.4.2, the OSI Reference Model places constraints on the use of

connectionless functions at the various layers. In particular, the OSI Reference Model prohibits the mapping between the connectionless-mode and the connection-mode except at the Transport/Network layer boundary and in the Application layer. Therefore, we can consider the use of the connectionless-mode at the lower layers (Transport layer and below) in this subclause distinct from (but complementary to) the use of the same mode in the upper layers (see 6.4.2).

7.3.3.1 Connectionless-mode Transport layer

By design, the connectionless-mode Transport protocol is restricted to providing very little functionality. It provides a protocol identifier, a protocol version and addressing selectors. The connectionless protocol overhead is as small as can be; so it does not warrant the creation of a special "efficient" version of this protocol.

In the case where more than one Application Entity (AE) needs to be addressed at the same time, it will be necessary to populate the transport selectors for the connectionless-mode protocol. This adds several octets to the Transport Unit-data PDU.

7.3.3.2 Connectionless-mode Network layer

The Connectionless-mode Network Layer Protocol (CLNP) is defined in ITU-T Rec. X.233 | ISO/IEC 8473-1 and can also be used without any special "efficiency" enhancements. There are two cases of use of the CLNP that must be considered:

1) If CLNP is used over a single subnetwork, i.e. addressing for the Network layer has the same scope as the addressing for the underlying Data Link layer, then the null encoding of the CLNP can be used. This consists of a single octet with the value "0" in from of all PDUs.

   NOTE 1 – With the use of this option, it is unclear why a connection-mode Network Layer Fast Byte protocol is required, since the latter can only be used under the same circumstances, i.e. over a single subnetwork, and results in a saving of only one octet per data PDU. Such savings can only be significant with very small PDU sizes in the order of 10 octets or less.

2) If the CLNP is used over multiple subnetworks, i.e. it has an addressing scope greater than any one of the underlying Data Link layers, then the full CLNP will be required. In such a case, there are some steps that can be taken to improve the efficiency. For example, the use of optional facilities and the use of fragmentation/reassembly should be avoided. Furthermore, the variable length of the NSAP address can be used to advantage. In many such uses, communication can be divided into two categories: internal communication within the enterprise or corporation and communication external to the enterprise or corporation. In many cases, the bulk of the communication will belong to the former category, not the latter. In such cases, the following approach, known colloquially as "4 digit dialling" can be taken to further enhance the efficiency of CLNP:

   a) Assign NSAPs according to a global naming tree. (These will be used for external communications.) The NSAP format will consist of an initial string (call it X) that is required to make the address unambiguous in the global naming space.

      NOTE 2 – Many NSAP addressing schemes include considerable political or administrative information. To the degree possible, this information should be eliminated (for efficiency purposes) and the NSAP address should utilize topological information only.

   b) At some point in the NSAP, there will be an identifier for the enterprise or the corporation. Call this Y. The NSAP address below this point will be unambiguous within the enterprise or corporation. Assign the rest of the NSAP address below this enterprise (or corporation) identifier to be topologically significant within the enterprise. Call this portion Z.

   c) Now each NSAP can be constructed as a string XYZ.

   d) For communication external to the enterprise or corporation, NSAPs of the form XYZ are used in the CLNP addressing fields. For communication within the enterprise or corporation, NSAPs with a local AFI followed by Z are used. Depending on the point of view, either we are assigning each device two addresses, one that looks like XYZ and the other that looks like a local AFI followed by Z, or we are only "dialling the last four digits". In either case, we may reduce the overhead of using CLNP by as much as 30 octets.

7.3.3.3 Connectionless-mode Data Link protocol

Data Link protocols are, by definition, media-specific and are highly optimized for the intended media. Again special "efficiency" measures are seldom applicable and generally cannot be made without jeopardizing the error characteristics and QoS required by the application. Once again, the same general rules applied at the Network layer (see 7.3.3.2) apply here, namely, avoid the use of optional facilities and, if possible, avoid fragmentation/reassembly.

7.4 Matching efficiency needs to various lower layer protocol options

7.4.1 Single subnetworks

7.4.1.1 Use of lower layer fast bytes

In matching efficiency needs to lower layer protocol options, there is a first trade-off between complete generality of addressing information and savings to be gained by prior knowledge of restricted topologies.

The minimal connectionless lower layers offer null encoding of the network and data link layers. This is feasible in this case because the network layer has the same scope as the data link layer. This is useful in configurations such as a single satellite link.

In matching efficiency needs, there is a second trade-off between complete generality of error control and savings to be gained by prior knowledge of error characteristics of the underlying subnetwork. In the case where the underlying subnetwork provides error control, enhancement to the underlying QoS is not required, and the transport protocol need only provide addressing information. In this case, consideration is given to the connectionless transport protocol with default values for the parameters.

7.4.2 Multiple subnetworks

7.4.2.1 Minimal connectionless lower layers

In matching efficiency needs to lower layer protocol options, there is a first trade-off between complete generality of addressing information and savings to be gained by prior knowledge of restricted topologies. When multiple subnetworks are employed, the above restriction must be relaxed.

The null encoding option of CLNP may be employed where enhancement of QoS is not required, and the reduction of round trip delay is paramount. This requires that addressing and QoS be implemented at the data link layer. The network may also be carefully architected such that the subnetwork capacities are not arbitrary. In such a case, routing in CLNP without the use of fragmentation is possible, since arbitrarily small pipes will not be encountered.

7.4.2.2 Minimal connectionless network layer and transport protocol class 4

A careful consideration of the network topology may allow the network paths to be fixed such that arbitrary routes and capacities do not have to be accommodated, mostly obviating network routing. A capable transport layer may also resequence PDUs, allowing further optimizations at the network layer. Such a configuration would require no PCI at the network layer.

7.4.2.3 Connection-oriented lower layers

The fast byte protocols for the connection-oriented lower layers define a single PDU to standardize the mapping between the transport, network, and data link services. Thus, these protocols are to be used in situations (e.g. satellite links) where the network topology is tightly constrained and the data link QoS is known and acceptable. The fast byte protocols eliminate the set-up round trips, but this can only occur where the topology is known by prior agreement.

8 Integration of efficiency options across the layers

8.1 Possible combinations of upper and lower layer efficiency options

In considering possible combination of upper and lower layer efficiency enhancements, two considerations are paramount.

At the upper layers, the needed generality of semantics must be considered. If presentation context must be negotiated, then complete statement of transfer syntax and concrete syntax is required. If, on the other hand, this is known to the communicating applications, then the entire context definition list need not be sent at every connection set-up.

At the lower layers, the needed generality of topology must be considered. If the topology is restricted, for example, by the specific subnetworks deployed, then the lower layer fast bytes may be included.

Generally, consideration of the domain requirements for the nominal functionality provided by each layer is the key to the decision to elide that layer. A domain may employ upper and layer efficiency options if application requirements are known, network topology is limited, and quality of service is acceptable.

The use of connectionless upper layers over connection-oriented lower layers is also a valuable tactic. The lower layers can provide addressing and quality of service enhancement while the upper layers provide minimal support.

8.2 Relaying issues

The use of relaying in a restricted topology may be possible while still using the efficiency enhancements. In the case of relaying, any link of the relaying system may be made more efficient if local knowledge can restore addressing or presentation context information. A relay itself may be made more efficient with prior knowledge of the input topology. This may be possible, for example, where a bandwidth-limited environment (e.g. air-ground) is conjoined to a less restricted environment (e.g. ground-ground). The bandwidth-limited environment may be restricted in terms of addressing and connection set-up until the point of entry into the less restricted environments. In such a situation, significant savings are possible with the use of all efficiency enhancement in the bandwidth-limited situation being conjoined with full addressing usage in the less constrained environment.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks and open system communications**
Series Y	Global information infrastructure
Series Z	Programming languages