

CONTENTS

Contents i

List of Figures ii

1. INTRODUCTION 1

2. PURPOSE AND SCOPE 1

 2.1 Definition of Transaction Capabilities. 1

 2.2 Scope of Transaction Capabilities. 1

 2.3 Scope of the Initial Specification of Transaction Capabilities. 1

3. ARCHITECTURAL CONCEPTS AND TERMINOLOGY 2

 3.1 Application of OSI Reference Model. 2

 3.2 Considerations. 2

4. OVERVIEW OF TC FUNCTIONS AND PROCEDURES 4

 4.1 Framework for Transaction Capabilities Protocol. 4

 4.2 Discussion. 4

 4.3 Identifying Services Required of Each Layer. 5

 4.4 General Description of TCAP Procedures. 6

 4.5 General Component Procedures. 7

 4.6 Service Procedures. 8

5. LAYER SERVICE CHARACTERISTICS 8

 5.1 Layer Services Assumed from the SCCP. 8

 5.2 Primitives and Layer Services for ASP Layers. 8

 5.3 Layer Services Provided to the Application Process. 8

6. STRUCTURE OF RECOMMENDATIONS 9

LIST OF FIGURES

| | |
|---|----|
| Figure 1/Q.771 - Transaction Capabilities Protocol Architecture | 3 |
| Figure 2/Q.771 - Identifying Layers and Services with Headers | 6 |
| Figure 3/Q.771 - State Diagrams for Components | 7 |
| Figure 4/Q.771 - Connectionless Transaction Primitives | 10 |

1. INTRODUCTION

This series of Recommendations specifies SS7 Transaction Capabilities or TC. The term "Transaction Capabilities" refers to the Application layer protocol, called Transaction Capabilities Application Part or TCAP, plus the supporting Presentation, Session and Transport layers, called the Application Service Part or ASP. To date, only SS7 MTP plus SCCP transport has been considered. Any standard OSI Network layer may be used in place of the MTP plus SCCP provided that performance requirements of the services being supported by the higher layers can be met. Other methods of transport are for further study.

2. PURPOSE AND SCOPE

2.1 Definition of Transaction Capabilities. Transaction Capabilities in the SS7 protocol are functions which control non-circuit-related information transfer between two or more signalling nodes via a signalling network.

2.2 Scope of Transaction Capabilities. Transaction Capabilities in a SS7 network should be considered for use between:

- 1) exchanges
- 2) exchanges and network service centers (e.g. data bases, service control points, OA&M centers)
- 3) subscribers and network service centers (in conjunction with the subscriber access protocol, e.g. Q.931)
- 4) subscribers (in conjunction with the subscriber access protocol, e.g. Q.931)

Furthermore, Transaction Capabilities in a SS7 network may interwork with transaction oriented information transfer originated in or destined for networks using other data communications protocols.

Transaction Capabilities provides a set of procedures which can be used for a variety of services, thereby avoiding the inefficiency of creating specific procedures tailored to a particular need. Thus, Transaction Capabilities provides a framework for a common approach to new services within a network as well as a framework for service architecture for cooperative inter-network services.

Wherever possible, procedures and formats for TC are based on existing CCITT Recommendations. The tangible benefits of such an approach are rapid documentation, reduced standardization effort, and faster and more widespread implementation (with resulting economies of scale and open environment for suppliers).

This approach is not intended to constrain any implementation of a service, whether it be intra- or inter-network. Similarly, no service or network architecture requirements are levied.

2.3 Scope of the Initial Specification of Transaction Capabilities. This initial specification is intended to provide, in an open-ended manner, the capabilities needed to support present and near-term services requiring transactions among exchanges, service control points, and data bases. Extension to other cases should be straightforward within the framework of Transaction Capabilities.

This initial specification addresses TC procedures relying on a connectionless network service. Procedures relying on a connection-oriented network service are for further study.

An asterisk "" indicates a change from the CCITT Red Book, Vol. VI, that is specific to U.S. Networks.

A bar "̄" indicates a change from Issue 1 of Bell Communications Research Specification of Signalling System Number 7, Vol. 1 and 2.

The specification itself is general in nature. Some inter-network services are seen to have sufficient interest across multiple networks that common agreements as to how they will operate across network boundaries are seen as desirable. These services are illustrated in annexes to Q.774, using the operations, parameters and procedures defined in Recommendations Q.772, Q.773 and Q.774 together with service-specific operations and parameters as required.

3. ARCHITECTURAL CONCEPTS AND TERMINOLOGY

3.1 Application of OSI Reference Model. The layered reference model is recognized as a useful tool in developing protocol specifications. From an end user point of view, Transaction Capabilities for initially planned services lie within the network layer of the OSI model. From the point of view of, for example, an exchange querying a data base, the exchange and the data base may also be seen as "end users". The nature of anticipated services to be supported suggests that the protocol may require the equivalent of many of the functions provided in OSI layers 6, 5 and 4, called the ASP in SS7 (Application Service Part consisting of Presentation, Session and Transport layers). Hence, there should be advantages in adopting these concepts to a protocol for network transactions.

Processes providing a transaction oriented service may appear at more than one signalling point. The architecture must be able to support global addressing of a transaction service and provide the functions needed to route signals to the appropriate points. The architecture must also provide management functions to handle congestion and failure of transaction processes.

Figure 1/Q.771 illustrates two ways in which the architecture of Transaction Capabilities may be modelled. The first (designated "a" in the figure) shows separate ASP entities for each process supported. The second (designated "b" in the figure) shows a common TCAP and ASP supporting more than one transaction process.

3.2 Considerations.

3.2.1 Addressing of Upper Layer Entities. The model uses the Sub-System Number (SSN) at the SCCP layer to identify the particular process. This allows the SCCP Global Title translation functions to be used to support global addressing of transaction services, in addition to point code and sub-system number addressing.

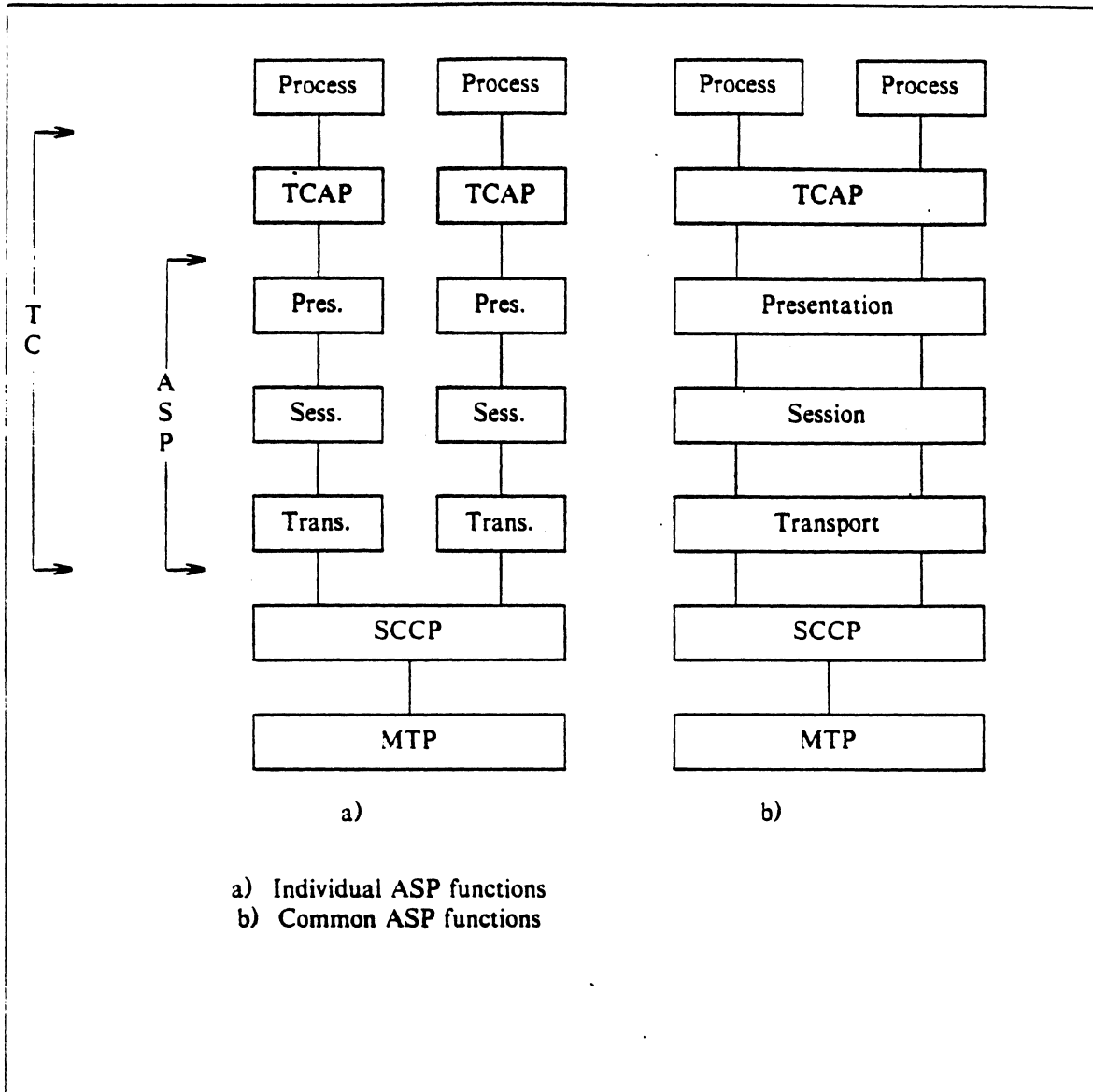


Figure 1/Q.771 - Transaction Capabilities Protocol Architecture

3.2.2 Management of Upper Layer Entities. Use of SSNs allows the SCCP Management procedures to be used to handle independently failing or congesting Application Processes.

3.2.3 Layered vs. Non-Layered. Some transaction-oriented services may not require any functions of the ASP. This suggests a non-layered approach may be preferable. However, as new transaction-oriented services are defined, a non-layered approach may complicate the protocol design. As well, with a non-layered approach, the benefits of commonality with other protocols could be lost. Transaction Capabilities have therefore been specified so as to readily allow inclusion of ASP functions when required. However, which ASP functions and how they should be included are for further study.

Since not all defined Presentation, Session and Transport functions are likely to be required at any one node in the network, it will not be necessary to provide all the possible ASP functions at every node. Suitable classes, options and functional units should be selected for each node as the needs of that node dictate. These can be expanded as required.

3.2.4 Architecture Model vs. Implementation. From the OSI point of view, a particular process has its own Application, Presentation, Session and Transport layers. This is illustrated in Figure 1/Q.771, part "a". Each "stack" is independent of the other "stacks". This concept is applied to the Transaction Capabilities protocol. Note that there is an implicit agreement to use the MTP and SCCP as common elements in this structure.

Within any particular implementation, it is possible to implement each block shown independently or to combine blocks vertically or horizontally or both where desired. Thus Figure 1/Q.771, part "b" may represent an example of an implementation where all the blocks are combined horizontally, while retaining the protocol model of part "a". As stated in Section 2.2, the approach is not intended to constrain any implementation of a service, whether it be intra- or inter-network.

4. OVERVIEW OF TC FUNCTIONS AND PROCEDURES

4.1 Framework for Transaction Capabilities Protocol. The initial Transaction Capabilities protocol, consisting of TCAP and the ASP, is based on the following CCITT Recommendations:

| | |
|--------------|--------------|
| Application | X.409, X.410 |
| Presentation | X.409, X.410 |
| Session | X.225 |
| Transport | X.224 |

4.2 Discussion.

4.2.1 Application. Section 2 (Remote Operations) of X.410 (Message Handling Systems: Remote Operations and Reliable Transfer Server) provides the basis for TCAP for foreseen services. It provides four OPDUs (Operation Protocol Data Units):

| | |
|---------------|--|
| Invoke | {Invoke ID, Correlation ID ¹ , Operation, Argument} |
| Return Result | {Correlation ID ² , Result} |
| Return Error | {Correlation ID ² , Error, Parameter} |
| Reject | {Correlation ID ² , Problem ³ } |

The Invoke OPDU requests that an operation be performed. The Return Result OPDU reports the successful completion of an operation. The Return Error OPDU reports the unsuccessful completion of an operation. The Reject OPDU reports the receipt and rejection of a incorrect OPDU. These OPDUs may be viewed as tools for constructing a Transaction Capabilities Application Protocol.

1 In TCAP, X.410 Remote Operations has been extended to include an optional ID to allow correlation of Invoke OPDUs to other Invoke OPDUs.
 2 Section 2/X.410 calls this ID the Invoke ID, it being understood that it is the reflection of the Invoke ID appearing in the Invoke OPDU. For TCAP, it is renamed Correlation ID for the OPDUs noted.
 3 In TCAP, X.410 Remote Operations has been extended to include a mandatory parameter for the Reject OPDU.

An OPDU as extended for TCAP is referred to as a Component. One or more Components may be carried in TCAP message.

4.2.2 Presentation. The purpose of the presentation layer is to provide functions to negotiate and select a transfer syntax and carry out transfer syntax transformations. These capabilities are not required for presently envisaged services.

The Components discussed in Section 4.2.1 above (i.e., the Presentation syntax of X.410) is all that is initially required. The use of Presentation layer services for future TC-supported services is for further study as such services and their presentation needs are defined.

4.2.3 Session. Initial services to be supported using Transaction Capabilities are not expected to require any session layer services. Hence, TC may operate in a connectionless mode in this case.

Services being considered for implementation in the medium to long term are likely to require session layer services. In this case, TC will operate in a connection-oriented mode since the session layer protocol assumes an underlying connection-oriented network.

X.225 specifies the session layer protocol. This protocol includes a wide range of capabilities. An initial subset of the OSI session protocol has been selected as likely needed to support future services using Transaction Capabilities since not all session layer capabilities are likely required for any one service.

A number of Functional Units (FUs) have been defined in the session layer protocol. The ones which may be applicable to Transaction Capabilities are the kernel FU (including the functionality for connection establishment, orderly release, abort and normal data transfer), duplex FU, half duplex FU, activity management FU, exceptions FU and minor synchronize FU.

The kernel and duplex FUs provide the Basic Combined Subset (BCS) of X.225 and are likely suitable for "short" connection-oriented transactions. The kernel plus the Basic Activity Subset of X.225 (activity management, exceptions, minor synchronize and half duplex FUs) are likely suitable for "long" transactions such as a file transfer service requiring high reliability.

This area is for further study as new services to be supported by Transaction Capabilities are identified as requiring session layer services.

4.2.4 Transport. Transport Class 0 service (Simple, i.e. no enhancements to the service provided by the network layer) should be appropriate since, for many transaction-oriented services, the Signalling System No. 7 network layer (provided by the MTP and SCCP) will provide a high degree of reliability. Hence, the transport layer would contain effectively null functionality to support presently foreseen transaction-oriented services.

The need for transport layer services required for new services to be supported by Transaction Capabilities is for further study.

4.3 Identifying Services Required of Each Layer. Figure 2/Q.771 illustrates the general concept of how services of the various layers are activated. As a message passes down through the layers, each layer places a header on it specific to the functions that layer performs. These functions may be grouped into classes or functional sets. The original Protocol Data Unit (PDU) plus headers of higher layers are treated as user information and are not examined by lower layers as the message is passed down. Similarly, when a message is being passed to a higher layer, each layer strips off its header before passing the message up as user data using the appropriate primitives. Some of the information contained in the headers will be preserved as parameters in the inter-layer primitives, for example, the called and calling addresses.

For some transaction services, both initially and in the future, some layers will be null. If a particular transaction service does not require presentation, session or transport layer services, then layer protocol headers are not required. This is the case when a connectionless mode of operation is used.

4.4 General Description of TCAP Procedures.

4.4.1 Types of Transactions. Transactions can be viewed at two levels: application process to application process, TCAP to TCAP. An application process transaction consists of one or more TCAP transactions. A TCAP transaction consists of one or more TCAP messages. TCAP transactions may be one-way (Unidirectional), two-way simple (single Query and Response), or two-way Conversational (multiple queries and/or multiple responses) in nature. The node initiating the transaction indicates which case applies from its point of view at the start of the transaction. The nature of a transaction may change during its life.

4.4.2 Initiation of Transactions. An Application Process Transaction is initiated with the assignment of a Transaction ID. A TCAP Transaction is initiated by sending or receiving an initiating TCAP message.

4.4.3 Termination of Transactions. An Application Process Transaction is terminated with the release of the Transaction ID. A TCAP Transaction is terminated by either a terminating TCAP message or at the discretion of the Application Processes at both ends (without an explicit message being sent) by informing their respective TCAPs..

4.4.4 Association of Application Process Transactions. Application Process Transactions are identified through Transaction IDs. These are carried in the TCAP message. Both Application Processes involved in the TCAP Transaction may assign Transaction IDs in any manner convenient to them.

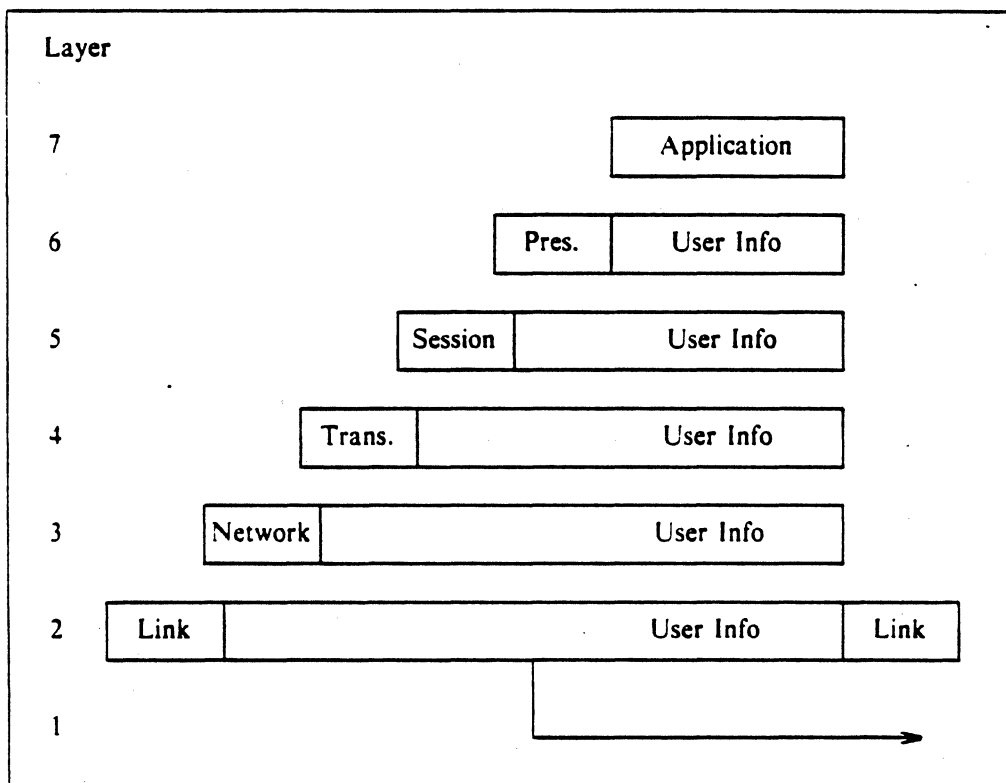


Figure 2/Q.771 - Identifying Layers and Services with Headers

4.4.5 Correlation of Components Within a single transaction, one or more operations may take place. For each operation, one or more Components may be involved.

Return Result, Return Error and Reject Components are correlated with Invoke Components through a Correlation ID which is set equal to the Invoke ID appearing in the Invoke Component (see Section 4.2.1). In the case where an Invoke Component must refer to another Invoke Component, it also contains a Correlation ID (set equal to the Invoke ID of the Invoke Component to which it needs to be correlated) as well as its own Invoke ID.

4.5 General Component Procedures. Depending on the service being supported, an invoked operation may report success or failure, failure only, success only, or neither. Figure 3/Q.771 contains an overview state diagram for an Invoke Component that succeeds and reports both success and failure. The state transition diagrams in Q.774 provide further details.

4.5.1 Operation Succeeds. When an originating node invokes an operation at a remote node, and success is reported, a Return Result or Invoke Component is sent.

4.5.2 Operation Fails. When no protocol error exists, and the operation invoked is not able to reach a successful result, a Return Error Component is sent indicating the reason for failure. This applies when the operation reports failure.

4.5.3 Protocol Error at Component Level. When a Component other than a Reject Component is incorrect, its receipt and rejection is reported using a Reject Component. Causes for rejection may include undetected bit errors, badly structured Components, duplicate invocations (when applicable to the service being supported), and unrecognized operations.

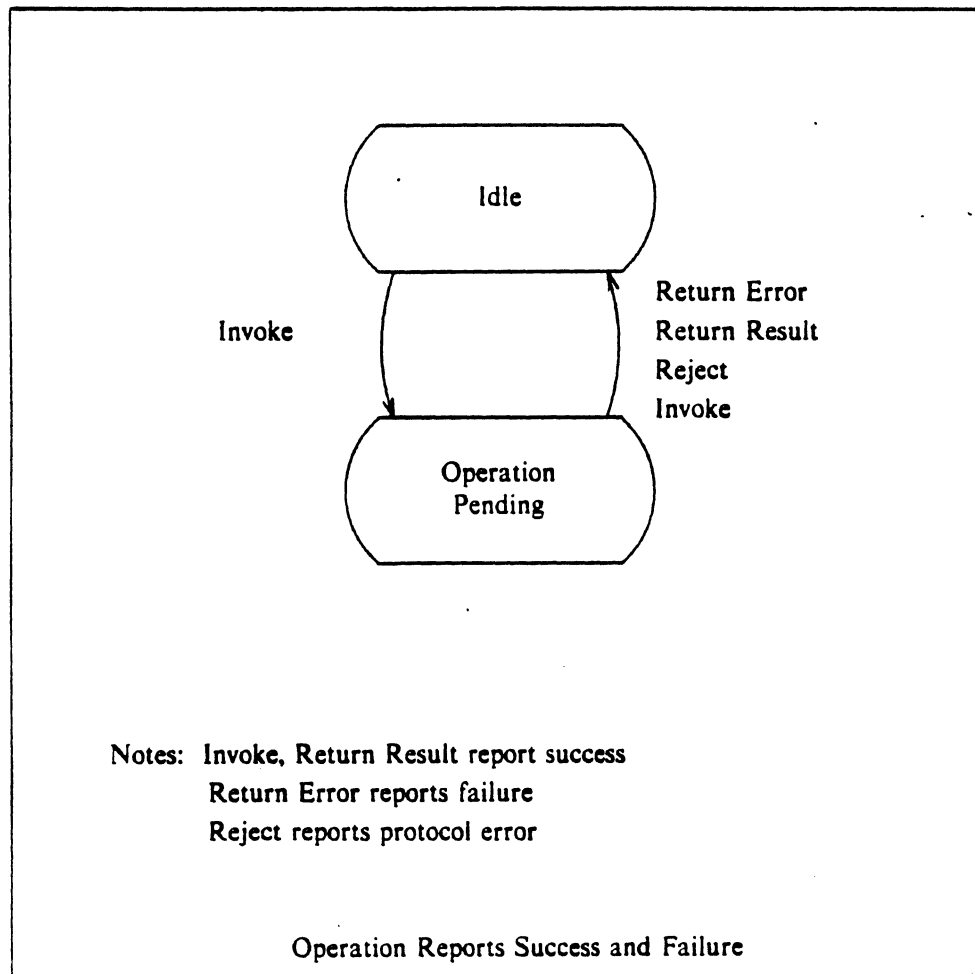


Figure 3/Q.771 - State Diagrams for Components

4.5.4 Protocol Error at TCAP Message Level.

When a malformed TCAP message is received (e.g., two Transaction IDs are indicated but only one is provided). A Reject Component is returned if the return address is available. It is coded to reflect an incorrect TCAP message as opposed to an incorrect component.

4.6 Service Procedures. Service procedures are defined using Components in sequences appropriate to the service being supported. Operations and parameters are defined as required for the service, including error conditions within the service. Such error conditions are not protocol errors but are part of the service definition. An example of such an error is incomplete customer input.

A set of operations and parameters are provided in Q.772 for use in developing procedures for specific services. Operations and parameters not included may be defined for the service in question as required, following the pattern of those provided.

5. LAYER SERVICE CHARACTERISTICS

5.1 Layer Services Assumed from the SCCP.

5.1.1 Description. The initial services supported by Transaction Capabilities require only SCCP Class 0 (basic connectionless service) or SCCP Class 1 (sequenced (MTP) connectionless service). Connection-oriented SCCP Classes 2 through 4 will be required for anticipated transaction-oriented services in the medium to long term. (Note that SCCP service classes and service classes of other layers do not have a one-to-one correspondence.)

5.1.2 Primitives and Parameters. The primitives and parameters between the SCCP and a user of the SCCP are described in section 2.2.2/Q.711 and Table 8/Q.711. Of these, N-UNITDATA will be required to support the initial set of transaction services.

The primitives N-NOTICE, N-COORD, N-STATE and N-TRAFFIC pass between SCCP management and the user of the SCCP. They are related to the management of service resources in specific network architectures. Their use for Transaction Capabilities is for further study.

5.2 Primitives and Layer Services for ASP Layers. Primitives for connectionless operation of the ASP are needed. T-, S- and P-UNITDATA primitives are shown in Figure 4/Q.771 with the parameters for N-UNITDATA (defined in Table 8/Q.711), namely Called Party Address, Calling Party Address, Quality of Service Parameter Set and User Data.

These primitives require no functionality at the indicated layers but preserve the layered structure of the protocol for future, more complex layer services. These more complex services will require primitives appropriate to the layer services requested of the ASP layers. These primitives are defined in the CCITT Recommendations listed in section 4.1 above.

Primitives for management purposes in a connectionless mode of operation analogous to N-NOTICE, N-COORD, etc. are for further study.

Each layer of the architecture introduces its own Protocol Data Units (PDUs) when required. Each PDU is carried as user data by lower layers. Thus, a presentation PDU or PPDU carries as user data the application PDU or APDU. The SPDU carries the PPDU as user data, etc. This applies when services of the ASP are used by TCAP.

When a connectionless mode of operation is used, no ASP services are required, therefore no fields in UNITDATA messages are allocated for transport, session or presentation PDUs.

5.3 Layer Services Provided to the Application Process. Transaction Capabilities provides the means for an Application Process at a given node to access Application Processes at other nodes via the SS7 network. Transaction Capabilities consists of a set of service elements each of which accepts and processes requests for provision of some Transaction Capabilities from the Application Process, or provides to the Application

Process some response as a result of a stimulus from Transaction Capabilities.

The application layer is responsible for the transfer of information between Application Processes. It is the highest layer of the OSI Reference Model and provides the sole means for application processes to access the capabilities available from the open systems interconnect environment.

Users of the application layer services are Application Processes. Application Process data is data transferred between two application entities on behalf of the Application Processes for whom the application entities are providing services. Application Process data consist of one or more components. A set of parameters which are associated with or required for the performance of processing functions is included in each Component.

In TCAP, an Application Protocol Data Unit (APDU) is a unit of data which contains one or more Components as described in section 4.2.1 above.

6. STRUCTURE OF RECOMMENDATIONS

Recommendation Q.772 defines message fields, components, operations, and parameters used within Transaction Capabilities. Message formats and codings are specified in Recommendation Q.773. TCAP procedures are specified in Recommendation Q.774.

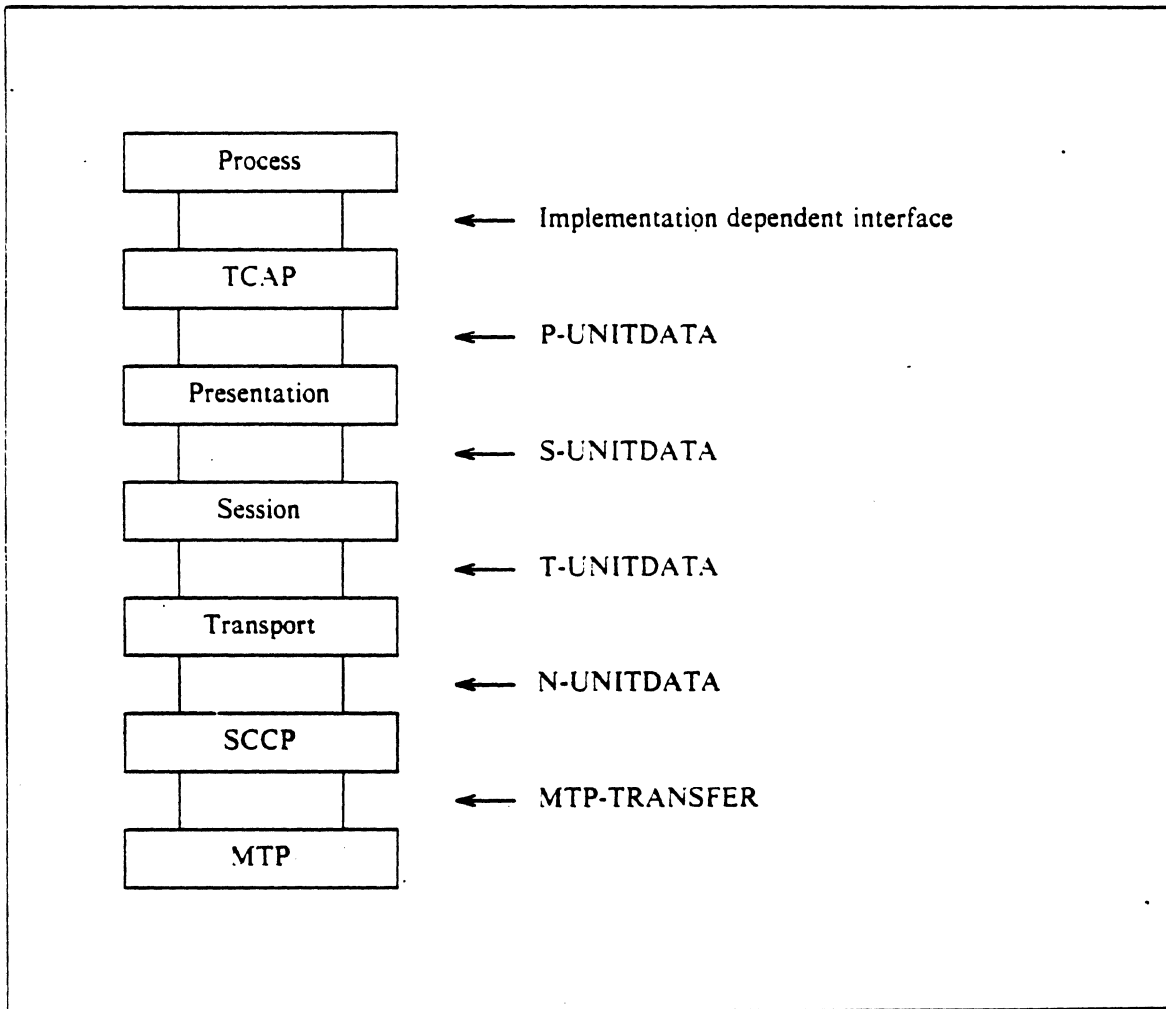


Figure 4/Q.771 - Connectionless Transaction Primitives

The recommendations include operations, parameters and procedures viewed as common to several services. *

The Annexes to Q.774 define how the operations, parameters and procedures specified in Q.772-Q.774 are used to support inter-network services agreed to be of common interest to many networks. These may be used and extended to develop new services as required. *

Where a service requires functionality of the supporting ASP layers, such services are cross-referenced to the appropriate CCITT Recommendations as listed in section 4.1. *