A standard session protocol for open systems interconnection (OSI)

by CHARLES E. YOUNG

*American Bell*
Lincroft, New Jersey

---

**ABSTRACT**

Protocol and service standards governing the session layer of the seven-layer reference model for OSI are expected to be finalized in 1983. These standards are being developed jointly by the International Organization for Standardization (ISO) and the International Telegraph and Telephone Consultative Committee (CCITT) groups, and will have wide applicability in data communications. Their basic characteristics and the open issues now being resolved are described. A status update on this fast-moving project will be presented at NCC '83.
INTRODUCTION

The Open Systems Interconnection (OSI) Reference Model describes seven independent layers of protocol for data communications among open systems. In general terms, it defines the services that each layer is to provide for its users (entities in higher layers). Each layer uses services provided by the layer below it, and enhances those services by means of its own protocol to provide services to its users. The seven layers are:

- Application layer
- Presentation layer
- Session layer
- Transport layer
- Network layer
- Data link layer
- Physical layer

Two companion standards are to be adopted for each layer, one defining the services provided by the layer and the other specifying the layer protocol. The service definition standards have no conformance requirements, and are intended mainly for use in guiding the protocol standardization efforts and for tutorial purposes. The protocol specification standards, however, are intended to specify all requirements necessary to permit different entities in the same layer, but usually in different end systems, to communicate in a fully compatible manner.

This article describes those service and protocol aspects of the session layer that are reasonably stable, as of October 1982. The services are described first, which helps in understanding what the protocol is being designed to achieve. Then the status of the protocol design efforts is described, followed by an outline of the more important open issues. An update report on the status of these open issues will be presented at NCC '83.

SESSION SERVICES

The session services currently being studied by ISO and CCITT would permit two users to organize and control their dialogue through use of a session connection established for them by the Session Layer (the provider). Service primitives are defined for use in establishing, utilizing, and releasing session connections, and four general types are expected to be used. A confirmed service primitive consists of four information transfers: a request from User 1 to the provider; a resulting indication from the provider to User 2; a response from User 2 to the provider; and a confirmation from the provider back to User 1. A nonconfirmed service primitive includes only a request and an indication; a single-user service primitive includes only a request; and a provider-initiated service primitive includes only indications, one to each user.

The service primitives that are currently agreed to in ISO and CCITT and their purposes are listed below according to the session connection phases in which they would be used and their types.

Establishment Phase

S_CONNECT (confirmed) would be used to establish a session connection and to negotiate values and ranges of values for session parameters.

Data Exchange Phase

S_DATA (nonconfirmed) would be used to transfer units of user data called session service data units (SSDUs) in sequence.
S_EXPEDITED_DATA (nonconfirmed) would be used to transfer expedited SSDUs, which may bypass other SSDUs in transit.
S_QUARANTINE_DELIVER (single-user) would be used to release for delivery a set of SSDUs previously stored by the provider at the sending user's request.
S_QUARANTINE_CANCEL (single-user) would be used to cancel delivery of a set of SSDUs previously stored by the provider at the sending user's request.
S_EXCEPTION_REPORT (provider-initiated) would be used to report unusual conditions not specifically covered by other services.
S_TOKEN_GIVE (nonconfirmed) would be used to transfer ownership of one or more tokens. A token is a session service attribute limiting the authority to request certain services to the owner of the token.
S_TOKEN请您_TOKE (nonconfirmed) would be used to request ownership transfer of specified tokens.
S_SYNC_MINOR (confirmed) would be used to insert a minor synchronization point in the stream of S_DATA primitives being sent from one user to the other. There are “explicit” and “potential” minor synchronization points, which would have different rules governing their responses and confirmations.
S_SYNC_MAJOR (confirmed) would be used to insert a major synchronization point in both streams of S_DATA primitives being sent in the two directions. All minor and major synchronization points are identified by a single set of serial numbers incremented by the provider.
S_RESYNCHRONIZE (confirmed) would be used to re­negotiate the current synchronization point serial-number value and token ownerships. With a “restart” resynchronization, the new serial number may be set back, but no lower than the most recent major synchronization point. Alternatively, the users may agree to abandon the old serial-number sequence and assign any new serial-number value for future use.

S_RESELECT (confirmed) would be used to revise parameter values and/or token ownerships, within ranges previously set during connection establishment.

S_ACTIVITY (nonconfirmed) would be used to initiate a new association between two users or to resume a specific previously unfinished association.

Release Phase

S_RELEASE (confirmed) would be used to release a session connection in an orderly manner without data loss. The release may be negotiated between the users in certain cases. S_U_ABORT (confirmed) would be used by a user to abort a session connection. Data in transit may be purged.

S_P_ABORT (provider-initiated) would be used by the provider to abort a session connection for reasons internal to the provider. Data in transit may be purged.

TABLE I—Service subsets

<table>
<thead>
<tr>
<th>Basic Subsets:</th>
<th>Half-Inter-Duplex</th>
<th>Half-Inter-Synchronous</th>
<th>Asymmetric-Synchronised</th>
<th>Asymmetric-Synchronised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Primitives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_CONNECT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_DATA</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_EXPEDITED_DATA</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_QUARANTINE_REDELIVER</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_QUARANTINE_CANCELLATION</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_EXCEPTION_REPORT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_TOKEN_GIVE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_TOKEN_REQUEST</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_TOKEN_REQUEST</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_ABORT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_RESYNCHRONIZE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_SELECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_ACTIVITY</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_RELEASE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_U_ABORT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S_P_ABORT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

TABLE II—Token availability definitions

<table>
<thead>
<tr>
<th>Token</th>
<th>Half-duplex data transfers (S_DATA)</th>
<th>S_SYNC_MAJOR</th>
<th>S_SYNC_MINOR</th>
<th>S_ABORT</th>
<th>S_RELEASE</th>
<th>S_CONNECT</th>
<th>S_DATA</th>
<th>S_EXPEDITED_DATA</th>
<th>S_QUARANTINE_REQUIREMENT</th>
<th>S_EXCEPTION_REPORT</th>
<th>S_TOKEN_GIVE</th>
<th>S_TOKEN_REQUEST</th>
<th>S_ABORT</th>
<th>S_RESYNCHRONIZE</th>
<th>S_SELECT</th>
<th>S_ACTIVITY</th>
<th>S_RELEASE</th>
<th>S_U_ABORT</th>
<th>S_P_ABORT</th>
<th>S_ANY_ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>send data</td>
<td>Half-duplex data transfers (S_DATA)</td>
<td>S_SYNC_MAJOR</td>
<td>S_SYNC_MINOR</td>
<td>S_ABORT</td>
<td>S_RELEASE</td>
<td>S_CONNECT</td>
<td>S_DATA</td>
<td>S_EXPEDITED_DATA</td>
<td>S_QUARANTINE_REQUIREMENT</td>
<td>S_EXCEPTION_REPORT</td>
<td>S_TOKEN_GIVE</td>
<td>S_TOKEN_REQUEST</td>
<td>S_ABORT</td>
<td>S_RESYNCHRONIZE</td>
<td>S_SELECT</td>
<td>S_ACTIVITY</td>
<td>S_RELEASE</td>
<td>S_U_ABORT</td>
<td>S_P_ABORT</td>
<td>S_ANY_ACTIVITY</td>
</tr>
<tr>
<td>synchronise minor</td>
<td>Minor synchronisation</td>
<td>No minor synchronisation</td>
<td>points permitted</td>
<td>points permitted</td>
<td>S_SYNC_MAJOR</td>
<td>S_SYNC_MINOR</td>
<td>S_ABORT</td>
<td>S_RELEASE</td>
<td>S_CONNECT</td>
<td>S_DATA</td>
<td>S_EXPEDITED_DATA</td>
<td>S_EXCEPTION_REPORT</td>
<td>S_TOKEN_GIVE</td>
<td>S_TOKEN_REQUEST</td>
<td>S_ABORT</td>
<td>S_RESYNCHRONIZE</td>
<td>S_SELECT</td>
<td>S_ACTIVITY</td>
<td>S_RELEASE</td>
<td>S_U_ABORT</td>
</tr>
<tr>
<td>synchronise major</td>
<td>Major synchronisation</td>
<td>No major synchronisation</td>
<td>points permitted</td>
<td>points permitted</td>
<td>S_SYNC_MAJOR</td>
<td>S_SYNC_MINOR</td>
<td>S_ABORT</td>
<td>S_RELEASE</td>
<td>S_CONNECT</td>
<td>S_DATA</td>
<td>S_EXPEDITED_DATA</td>
<td>S_EXCEPTION_REPORT</td>
<td>S_TOKEN_GIVE</td>
<td>S_TOKEN_REQUEST</td>
<td>S_ABORT</td>
<td>S_RESYNCHRONIZE</td>
<td>S_SELECT</td>
<td>S_ACTIVITY</td>
<td>S_RELEASE</td>
<td>S_U_ABORT</td>
</tr>
<tr>
<td>release session</td>
<td>Token assignment required</td>
<td>Only non-negotiated</td>
<td>to initiate orderly release</td>
<td>S_RELEASE</td>
<td>S_CONNECT</td>
<td>S_DATA</td>
<td>S_EXPEDITED_DATA</td>
<td>S_EXCEPTION_REPORT</td>
<td>S_TOKEN_GIVE</td>
<td>S_TOKEN_REQUEST</td>
<td>S_ABORT</td>
<td>S_RESYNCHRONIZE</td>
<td>S_SELECT</td>
<td>S_ACTIVITY</td>
<td>S_RELEASE</td>
<td>S_U_ABORT</td>
<td>S_P_ABORT</td>
<td>S_ANY_ACTIVITY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CONTROL OF SERVICE USAGE

The specific, detailed definition of each service primitive, its particular parameters and their permitted values and ranges of values, which are currently being studied in CCITT and ISO, will place inherent limits on the usage of every service primitive. Numerous modifications in these items will continue to be made until the protocol development efforts are completed. The current status of these items is contained in the most recent Draft Session Service Definition.3

Two other mechanisms, service subsets and tokens, are also included in the basic session services for purposes of usage control.

Service subsets are defined to match specific types of session users' needs, which are expected to predominate OSI session service usage. Five service subsets (see Table I) are currently defined, based largely on the session service primitives permitted to be used, but also containing certain detailed differences in parameter usage rules. Implementations only intended to provide services for specific user types can thus be simplified.

Four tokens are provided for use in selected subsets, as shown in Table I. Service subset selection during connection establishment determines which tokens are "available" and which ones are "not available" for use throughout the session. The particular meanings of these two terms for each token are shown in Table II. Table III lists the service primitives that can only be initiated by a user owning specific tokens.

SESSION PROTOCOL

The session protocol4 being developed will specify the rules for information exchanges between peer entities in the session layer. The units of information exchange are called session protocol data units (SPDUs). SPDUs will be used in various ways to transfer information in support of session service primitives. The relationships between SPDUs and the session service primitives are illustrated in Table IV, and the ways that
TABLE III—Token assignments required

<table>
<thead>
<tr>
<th>Service Primitive</th>
<th>To Be Initiated</th>
<th>Tokens Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_DATA send data*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_SYNC_MINOR send data* and synchronize minor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_SYNC_MAJOR send data*, synchronize minor and synchronize major</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_ACTIVITY send data* and synchronize major</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S_RELEASE send data* and release session*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Required only if "available."

SPDUs are currently proposed to be used in support of each service primitive are listed below.

It is important to recognize that the functions performed in the session protocol for a given service primitive are only permitted to take place when the associated service primitive is permitted to be used. Thus, the service subset and token authorization structures for the session services are reflected directly into the protocol, with respect to the functions permitted.

SPDU Usage for Each Service Primitive

S.CONNECT. Connect (CN) conveys connection request information from one session entity to the other. The normal response is either accept (AC) or refuse (RF). CN and AC contain information in parameters, such as the proposed subset and initial token assignments, which are negotiated among the two users and the two session entities to establish the characteristics and bounds permitted throughout the remainder of the session connection. These and all other SPDUs can only be conveyed via a transport connection, which must be provided by the transport layer. A characteristic of the initial session protocol is that only the session entity which previously requested the transport connection is permitted to send a CN SPDU over it.

S.DATA. Data transfer (DT) conveys session user data during the data transfer phase, which begins for a session entity when it sends or receives a valid AC. The right to send DT at any moment is governed by the send data token in half-duplex operation and by additional restrictions associated with synchronization and reselection functions when they are in use. Session connection flow control is attained only by refusing to accept further data transfers, and is reflected to the session sender through lower-layer protocol mechanisms. This technique is called backpressure.

S_EXPEDITED_DATA. Expedited (EX) conveys limited amounts of expedited user information using the transport Expedited service. EX SPDUs may bypass DT SPDUs enroute to the receiving user.

TABLE IV—Session protocol data units (SPDUs) for each service primitive

<table>
<thead>
<tr>
<th>Service</th>
<th>Data Unit</th>
<th>(SPDUs) Initiated By:</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_CONNECT</td>
<td>Connect (CN)</td>
<td>Accept (AC)</td>
</tr>
<tr>
<td></td>
<td>Data Transfer (DT)</td>
<td>Refuse (RF)</td>
</tr>
<tr>
<td>S_DATA</td>
<td>Expedited (EX)</td>
<td></td>
</tr>
<tr>
<td>S_SYNC_MINOR</td>
<td>Data Transfer (DT)*</td>
<td>Mark Confirmation (MC)</td>
</tr>
<tr>
<td>S_SYNC_MAJOR</td>
<td>Data Transfer (DT)*</td>
<td>Prepare (PR)</td>
</tr>
<tr>
<td>S_RESynchronize</td>
<td>Prepare (PR)</td>
<td>Resynchronize</td>
</tr>
<tr>
<td></td>
<td>Resynchronize</td>
<td>Acknowledgement (RA)</td>
</tr>
<tr>
<td>S_RERELEASE</td>
<td>Reselect (RE)</td>
<td>Acknowledgement (RA)</td>
</tr>
<tr>
<td>S_ACTIVITY</td>
<td>Start Activity (AA)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Continue Activity (CA)</td>
<td></td>
</tr>
<tr>
<td>S_RELEASE</td>
<td>Finish (FN)</td>
<td>Disconnect (DN)</td>
</tr>
<tr>
<td></td>
<td>Not Finished (NF)</td>
<td></td>
</tr>
<tr>
<td>S_ABORT</td>
<td>Abort (AB)</td>
<td>Abort Accept (AA)</td>
</tr>
<tr>
<td>S_P_ABORT</td>
<td>Abort (AB)*</td>
<td>Abort Accept (AA)*</td>
</tr>
</tbody>
</table>

*Service information is contained only in parameters of the SPDU.
**Provider initiated.

S_QUARANTINE_DELIVER. The end of quarantine parameter in DT is used to signal that data quarantined for delayed delivery, if any, should be released for delivery.

S_QUARANTINE_CANCEL. Cancel (CL) is used to signal that data quarantined for delayed delivery, if any, should be canceled.

S_EXCEPTION_REPORT. Exception (ER) is used to convey information concerning unusual conditions detected in the session layer, which should be reported to the distant session user.

S_TOKEN_GIVE. Either Give tokens (GT) or the give tokens parameter in DT (when there is data to send) may be used to signal transfer of tokens from the owning user to the other user.

S_TOKEN_PLEASE. Please tokens (PT) is used to signal a request to transfer tokens from the owning user to the requesting user.

S_SYNC_MINOR. Parameters in DT are used to signal a
potential or explicit minor synchronization point between two
SSDUs. Mark confirmation (MC) is used to acknowledge all
minor synchronization points up to one identified by its serial
number in a parameter of MC.

S_SYNC_MAJOR. Parameters in DT are used to signal a
major synchronization point between two SSDUs. Prepare
(PR), returned via the transport-expedited service, is trans­
ferred just prior to MC in response to receipt of a major
synchronization point. PR marks a major synchronization
point in the expedited-data path, and signals that the MC is
being sent. MC marks the point in the return normal data flow
of the major synchronization point, thus concluding the pre­
vious dialogue unit. (PR is only sent when the transport-
expedited service is in use).

S_RESYNCHRONIZE. Resynchronization is initiated by
transmission of PR, with a parameter specifying that re­
synchronize (RS) is being sent. RS is used to force re­
synchronization in the exchange of data to a specified syn­
chronization point and to force renegotiation of token own­
erships. Typically, resynchronization is utilized after a failure
is detected. Resynchronize acknowledgment (RA), preceded
by a PR message, is the normal response to RS. RA includes
the agreed new token ownerships and the new serial number
as parameters. Data may be lost as a result of the re­
synchronization function. (PR is only sent when the transport
expedited service is in use).

S_RESELECT. Reselect (RE) is sent to initiate renegotia­
tion of parameter values and/or token ownerships within
ranges established during session connection establishment.
The normal response is reselect acknowledgment (RK), com­
pleting the renegotiation.

S_ACTIVITY. Start activity (SA) signals the beginning of
a new association between two session users and conveys the
initial synchronization serial number for the association. It
also can convey a limited amount of user data. If the activity
type signaled in S_ACTIVITY is "resume," instead of
"start," then continue activity (CA) is sent, which must also
convey the identity of the previously unfinished user associa­
tion.

S_RELEASE. Orderly release is initiated by finish (FN).
The normal response is disconnect (DN); however, the re­
ceiving user is permitted to decline in certain cases when the
release session token is available. Not finished (NF) may then
be returned in response to FN.

S_U_ABORT and S_P_ABORT. Abort (AB) is used to
initiate the abort procedure, with the source (user or session
entity) and reason included as parameters. Abort accept
(AA) is the proper response. Parameters in AB and AA may
also be used to negotiate whether or not the same transport
connection may be reused to establish a new session con­
nection.

CURRENT PROTOCOL DESIGN STATUS

The session protocol for OSI is being designed taking into
consideration two existing standards, CCITT Recommend­
ation S.62\(^2\) and ECMA-75.\(^5\) Together these two standards
will permit the session services needed for OSI to be provided
in a manner that satisfies the identified needs of both CCITT
and ISO. S.62, having been designed prior to the OSI refer­
ce model, specifies some items clearly belonging in layers
above the session layer. These items will be handled by the
session protocol as user data. Also, some of the session func­
tions and many of the parameters are defined differently in
S.62 and in ECMA-75. The major current effort is to reconcile
these differences in a manner that satisfies the session service
definition and requires the minimum amount of change in
either standard.

**Major Issues and Status**

Several of the major issues identified initially have already
been successfully resolved, as of October 1982. It has been
agreed to use the S.62 encoding principles, and compatible
release and abort procedures have been agreed on, including
compatible rules governing reuse of an underlying transport
connection after completion of a session connection. Most of
the protocol data units in the two standards have been mapped
together; however, there are a few basic issues remaining,
which are currently being studied. Most of them involve syn­
chronization, and the more significant issues currently identi­
fied are outlined as follows:

1. Should the S.62 window mechanism, which permits mul­
tiple minor synchronization points to remain out­
standing, be controlled in the session protocol or left for
control in the users’ protocol? The alternatives result in
different session protocol mappings.

2. Should the users, when employing the session syn­
chronization services, be permitted to request services,
such as reversal of control states governed by tokens,
while outside of any user association known to the ses­
sion layer? The alternative is to require the users to enter
into a new activity before requesting such services,
which results in different protocol mappings.

3. In S.62 there is a “master/slave” relationship in the pro­
tocol, whereas the ECMA-75 protocol is symmetrical.
How should the master/slave relationship concept be
integrated into the overall protocol (without simply hav­ing
two different protocol machines)?

**REFERENCES**


2. “Reference Model Of Open Systems Interconnection For CCITT Applica­
tions,” COM VII—No. R9(C), Appendix 12 of Annex 8 to “Report Of The
Working Party VIII Melbourne Meeting, March 1982.”

3. “Draft Basic Connection-Oriented Session Service Definition,” ISO/TC97/
SC16 N1166, June 1982.

4. “Current Status of Draft Basic Connection-Oriented Session Protocol Speci­

5. “Control Procedures For The Telex Service,” CCITT Recommendation