The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

SPACE LINK EXTENSION—FORWARD CLTU SERVICE SPECIFICATION

DRAFT RECOMMENDED STANDARD

CCSDS 912.1-P-2.1

PINK SHEETS
December 2008

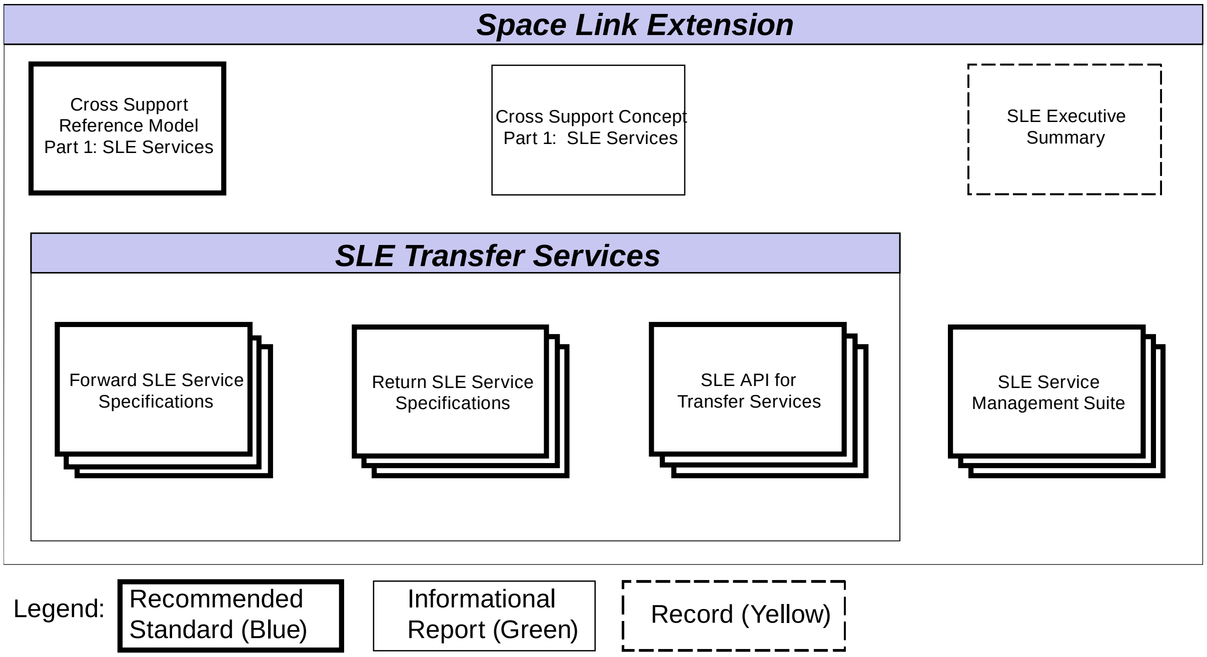
Space Link Extension
Cross Support Reference Model Part 1: SLE Services
Cross Support Concept Part 1: SLE Services
SLE Executive Summary
SLE Transfer Services
Forward SLE Service Specifications
Return SLE Service Specifications
SLE API for Transfer Services
SLE Service Management Suite
Legend:
Recommended Standard (Blue)
Informational Report (Green)
Record (Yellow)

Figure 1-1: SLE Services Documentation

a) *Cross Support Concept – Part 1: Space Link Extension Services* (reference [E3]): a Report introducing the concepts of cross support and SLE services;

b) *Cross Support Reference Model—Part 1: Space Link Extension Services* (reference [1]): a Recommended Standard that defines the framework and terminology for the specification of SLE services;

c) *SLE Service Management Specification Suite;* a set of Recommended Standards that establish the basis for SLE service management;

d) *Forward SLE Transfer Service Specifications;* a set of Recommended Standards that will provide specification of all forward link SLE transfer services (this Recommended Standard is one of the specifications in that set);

e) *SLE API for Transfer Services Specifications:* a set of Recommended Standards that provide specifications of an Application Program Interface and a mapping to TCP/IP as underlying communications service for SLE services;

f) *Return SLE Transfer Service Specifications;* a set of Recommended Standards that will provide specification of all return link SLE transfer services.

r) SLE Complex;

s) SLE Complex Management;

t) SLE data channel;

u) SLE functional group (SLE-FG);

v) SLE protocol data unit (SLE-PDU);

w) SLE service data unit (SLE-SDU);

x) SLE service package;

~~y) SLE System;~~

y) SLE transfer service instance;

z) SLE transfer service production;

aa) SLE transfer service provision;

bb) SLE Utilization Management;

cc) space link;

dd) space link data channel;

ee) space link data unit (SL-DU);

ff) space link session.

1.6.1.6 Additional Definitions

For the purposes of this Recommended Standard, the following definitions also apply.

1.6.1.6.1 Association

An association is a cooperative relationship between an SLE service-providing application entity and an SLE service-using application entity. An association is formed by the exchange of SLE protocol data units through use of an underlying communications service.

1.6.1.6.2 Communications Service

A communications service is a capability that enables an SLE service-providing application entity and an SLE service-using application entity to exchange information.

NOTE – If an SLE service user and an SLE service provider are implemented using different communications services, then interoperability between them is possible only by means of a suitable gateway. Adherence to this Recommended Standard ensures, at least in principle, that it is possible to construct such a gateway.

1.6.1.6.3 Confirmed Operation

A confirmed operation is an operation that requires the performer to return a report of its outcome to the invoker.

1.6.1.6.4 Initiator

The initiator is the object that issues the request to bind to another object (the responder).

NOTE – In other words, the initiator is always the invoker of the request to bind to another object. Therefore, in the context of the request to bind, the terms ‘initiator’ and ‘invoker’ refer to the same object and are synonyms.

1.6.1.6.5 Invocation

The invocation of an operation is the making of a request by an object (the invoker) to another object (the performer) to carry out the operation.

1.6.1.6.6 Parameter

A parameter of an operation is data that may accompany the operation’s invocation or return.

NOTE – The term parameter is also used to refer to mission-dependent configuration information used in production or provision of the service.

1.6.1.6.7 Performance

The performance of an operation is the carrying out of the operation by an object (the performer).

1.6.1.6.8 Port Identifier

A port identifier identifies a source or a destination in a communications system.

NOTE – See 2.6.4.6 for more information.

1.6.1.6.9 Responder

The responder is the object that receives a request to bind and completes the binding (if possible) with the initiator in order for a service association to exist between the two objects.

NOTE – In other words, the responder is always the performer of the binding. Therefore, in the context of binding, the terms ‘responder’ and ‘performer’ refer to the same object and are synonyms.

1.6.1.6.10 Return

The return of an operation is a report, from the performer to the invoker, of the outcome of the performance of the operation.

1.6.1.6.11 Service Instance Provision Period

A service instance provision period is the time during which a service instance (i.e., the capability to transfer one or more SLE data channels of a given type) is scheduled to be provided.

1.6.1.6.12 Unconfirmed Operation

An unconfirmed operation is an operation that does not require a report of its outcome to be returned to the invoker by the performer.

1.6.2 NOMENCLATURE

The following conventions apply throughout this Recommended Standard:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

1.6.3 CONVENTIONS

1.6.3.1 Specification of Operations

1.6.3.1.1 General

Section 3 of this Recommended Standard specifies the operations that constitute the Forward CLTU service. The specification of each operation is divided into subsections as follows:

needed to establish communications between the user and provider and to route SLE-PDUs between them. The initiator port identifier identifies the endpoint that will invoke the CLTU-BIND operation (initiator). The responder port identifier identifies the endpoint that will perform the CLTU-BIND operation (responder). Generally speaking, the information represented by a port identifier consists of:

a) information needed to route data between two real systems over a communications channel or network; and

b) information needed to route data within a real system to a particular application entity.

For example, the information represented by a port identifier might be the combination of an Internet Protocol (IP) network address and a Transmission Control Protocol (TCP) port number or the combination of an OSI network address and an associated set of service access points (SAPs).

The exact relationship between SLE port identifiers and communications ports provided by the underlying communications service must be specified by the mapping of the Forward CLTU service to the underlying communications service.

One possible mapping of the SLE transfer service to the TCP/IP communications service is specified in [E8]. As part of this mapping, also issues such as sizing of TCP buffers in accordance with the bandwidth-delay product of the communication link and ways to manage relative priority of transfer services concurrently using the same connectivity are to be addressed.

In order for an SLE association to be established, SLE Complex Management and SLE Utilization Management must agree beforehand on the responder port identifier for the association. The responder needs the information represented by the responder port identifier to ensure that resources are allocated to recognize and respond to a CLTU-BIND invocation for that association. The initiator needs the information to ensure that the CLTU-BIND invocation will be communicated to the appropriate responder.

In general, it is not necessary for SLE Complex Management and SLE Utilization Management to agree beforehand on the initiator port identifier for the association. Rather, the initiator should communicate that information to the responder in conjunction with the CLTU-BIND invocation. The exact means by which the initiator port identifier is provided to the responder is technology-specific and must be specified by the mapping of the Forward CLTU service to the underlying communications service.

The responder port identifier is included as a parameter of the CLTU-BIND operation. While it is only necessary that the SLE application entity communicate the information represented by the port identifiers to the underlying communications service, the responder port identifier is provided as a parameter of the CLTU-BIND operation to allow for the possible simplification of the implementation of a gateway.

3.1.5 AUTHENTICATION

NOTE – Requirements for security depend on the application and the ~~SLE system~~ environment of the SLE Complexes and the MDOS (e.g., whether closed or public networks are used or if access is only from physically restricted areas). In many environments, security may be provided by the communications service, transparently to the SLE application. This Recommended Standard does not preclude the use of security features that are provided by the communications service or the local environment, nor does it assume the availability of such features.

3.1.5.1 The Forward CLTU service shall provide the following options with respect to level of authentication of invocations and returns of operations:

a) ‘all’: all Forward CLTU invocations and returns, except the invocation of CLTU-PEER-ABORT, shall be authenticated;

b) ‘bind’: only the CLTU-BIND invocation and return shall be authenticated;

c) ‘none’: no Forward CLTU invocations or returns shall be authenticated.

3.1.5.2 SLE Complex Management and SLE Utilization Management shall agree on the level of authentication to be required for an association between a service user and a Forward CLTU service provider and shall configure both entities accordingly.

3.1.5.3 SLE Complex Management and SLE Utilization Management shall agree on the algorithm used to generate and check credentials parameters and make this algorithm known to the service user and service provider, together with associated parameters such as passwords or keys as necessary for the adopted algorithm.

NOTES

1 The specification of the algorithms themselves is outside the scope of this Recommended Standard.

2 The `initiator-identifier` and `responder-identifier` parameters of the CLTU-BIND operation identify the user and provider, respectively, and therefore the applicable authentication level and algorithm necessary to generate and check credentials.

3.1.5.4 For operations for which authentication of credentials is required by terms of the agreement between SLE Complex Management and SLE Utilization Management:

a) invocations shall include an `invoker-credentials` parameter to permit the performer to authenticate the invocation; and

b) returns shall include a `performer-credentials` parameter to permit the invoker to authenticate the return.

3.1.6.8 Compliance with this Recommended Standard does not require the performer to process invocations concurrently; however, the performer must accept invocations from a non-blocking invoker and buffer and serialize them by local means not visible externally.

3.1.7 TIME

3.1.7.1 The time reference for all parameters containing a time value shall be based on Coordinated Universal Time (UTC).

3.1.7.2 The type of parameters containing a time value shall be the CCSDS Day Segmented (CDS) time code format (reference [5]) with ~~a resolution of microseconds,~~ an epoch of 1958-01-01 and a 16-bit day segment. Depending on the F-CLTU service provider capabilities and/or the supported mission requirements, the time tag may have either a resolution of microseconds or a resolution of picoseconds.

3.1.7.3 All time values shall be expressed to a precision of at least one-tenth (0.1) of a second.

3.1.7.4 All time value shall be accurate to within one-tenth (0.1) of a second or better.

3.1.8 DELIVERY MODES

3.1.8.1 Forward Online Delivery

3.1.8.1.1 Forward online delivery service provision shall occur at the same time as service production, i.e., during a space link session.

3.1.8.1.2 CLTUs supplied by the service user shall be buffered by the service provider until they are processed.

3.1.8.1.3 The buffer used by the service provider exists only during service provisioning.

3.1.8.1.4 The timing of CLTU processing shall be determined by the order of CLTUs in the buffer and any annotation data provided with the CLTUs.

NOTE – The forward online delivery mode is defined in this Recommended Standard.

3.1.8.2 Forward Offline Delivery

3.1.8.2.1 Service provision and service production shall not overlap.

3.1.8.2.2 CLTUs supplied by the service user during service provision shall be buffered by the provider in persistent storage until service production.

NOTE – The forward offline delivery mode is outside the scope of this version of this Recommended Standard.

3.1.9 SETTING OF PARAMETERS

3.1.9.1 A Forward CLTU provider shall permit setting of the service configuration parameters as specified in table 3-1.

3.1.9.2 The range or set of values a parameter may assume is constrained by specification of its data type (see annex A).

3.1.9.3 Service management may further constrain the allowed values for a given service instance.

Table 3-1: Setting of Forward CLTU Service Configuration Parameters

Parameter	Service Management	CLTU-START Operation	CLTU-SCHEDULE-STATUS-REPORT Operation	CLTU-THROW-EVENT Operation (NOTE 3)
acquisition-sequence-length	X			
bit-lock-required	X			
clcw-physical-channel	X			
clcw-global-VCID	X			
delivery-mode	X			
expected-cltu-identification		X		
expected-event-invocation-identification				X
maximum-cltu-length	X			
maximum-reporting-cycle	X			
minimum-delay-time	X			
minimum-reporting-cycle	X			
modulation-frequency	X			
modulation-index	X			
notification-mode	X			
plop-1-idle-sequence-length	X			
plop-in-effect	X			
protocol-abort-mode	X			
reporting-cycle			X	

Parameter	Service Management	CLTU-START Operation	CLTU-SCHEDULE-STATUS-REPORT Operation	CLTU-THROW-EVENT Operation (NOTE 3)
return-timeout-period	X			
rf-available-required	X			
service-instance-provision-period	X			
service-version-number	X			
subcarrier-to-bit-rate-ratio	X			

NOTES

1 Further details on protocol-abort-mode are discussed in 4.1.5. The notification-mode parameter is described in 3.7.2.3. Other parameters are presented and described in table 3-11. A complete list of parameters that may affect service production is to be found in the service management suite of documents (see 1.5.2).

2 The user can ascertain the current value of the parameters presented in table 3-11 by means of the CLTU-GET-PARAMETER operation.

3 The ability to modify selected service configuration parameters using the CLTU-THROW-EVENT operation is allowed but not mandated in this Recommended Standard.

3.1.10 PROVIDER BUFFERING REQUIREMENTS

3.1.10.1 The service package shall specify the amount of buffering the provider must maintain.

3.1.10.2 The amount of buffer space shall be specified in terms of the number of octets that can be stored.

3.1.10.3 The service provider shall buffer only complete CLTUs.

3.1.11 ACCOUNTING SUMMARY

Statistical information to be collected over a period of time shall always refer to the service instance provision period.

3.2.2.8.2 `version-number` is conditionally present in the return based on the `result` parameter:

a) if the value of `result` is ‘positive result’, `version-number` shall be present in the return;

b) if the value of `result` is ‘negative result’, `version-number` shall not be present in the return.

3.2.2.8.3 If the value of `result` is ‘positive result’, the responder shall either:

a) accept the version proposed by the initiator by putting the same version number into the positive return; or

b) if the responder supports version negotiation, propose a lower (earlier) version number by putting the lower version number in the return.

3.2.2.8.4 If the responder implementation does not support the requested version and does not support a lower version (or does not support version negotiation), the responder shall reject the bind with the `diagnostic` parameter set to ‘version not supported’.

3.2.2.8.5 If the responder proposes a lower version in the return and the initiator does not support version negotiation or does not support the version proposed by the responder, the initiator shall unbind the association.

3.2.2.8.6 The `version-number` value of the Forward CLTU service defined by this issue of this Recommended Standard shall be ‘~~2~~3’.

NOTE – The version negotiation process as outlined above is only feasible as long as future versions of the Forward CLTU service retains the specification of the CLTU-BIND operation.

3.2.2.9 `service-instance-identifier`

The **`service-instance-identifier`** parameter shall uniquely identify this service instance within the scope of the service-providing SLE Complex.

3.2.2.10 `result`

3.2.2.10.1 The **`result`** parameter shall specify the result of the CLTU-BIND invocation and shall contain one of the following values:

a) ‘positive result’—the CLTU-BIND operation has been performed by the responder and the association is established;

3.6 CLTU-TRANSFER-DATA

3.6.1 PURPOSE

3.6.1.1 The user shall invoke the CLTU-TRANSFER-DATA operation to transfer CLTUs to the provider.

3.6.1.2 The provider shall provide a report of the outcome of the performance of the CLTU-TRANSFER-DATA operation to the user.

3.6.1.3 The CLTU-TRANSFER-DATA operation is valid only in state 3 ('active') and shall be invoked only by the user.

3.6.2 INVOCATION, RETURN, AND PARAMETERS

3.6.2.1 General

The parameters of the CLTU-TRANSFER-DATA operation shall be present in the invocation and return as specified in table 3-6.

Table 3-6: CLTU-TRANSFER-DATA Parameters

Parameters	Invocation	Return
`invoker-credentials`	M	
`performer-credentials`		M
`invoke-ID`	M	M
`cltu-identification`	M	M
`earliest-radiation-time`	M	
`latest-radiation-time`	M	
`delay-time`	M	
~~R~~report	M	
~~D~~data	M	
`cltu-buffer-available`		M
~~R~~result		M
~~D~~diagnostic		C

3.6.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the CLTU-TRANSFER-DATA invocation (see 3.1.5).

3.6.2.6 earliest-radiation-time

3.6.2.6.1 The **earliest-radiation-time** parameter shall be used to specify the earliest time that the provider shall start processing this CLTU.

3.6.2.6.2 The earliest-radiation-time parameter shall either be unspecified, i.e., contain a value of ‘null’, or shall specify the earliest time at which the leading edge of the first bit of the CLTU may begin to be radiated.

3.6.2.6.3 If the earliest-radiation-time parameter is unspecified, the provider shall begin processing immediately after any delay associated with the previous CLTU has timed out, as long as the production process is running (i.e., not ‘halted’ or ‘interrupted’).

3.6.2.7 latest-radiation-time

3.6.2.7.1 The **latest-radiation-time** parameter shall be used to specify the latest time at which the provider shall start processing this CLTU.

3.6.2.7.2 The latest-radiation-time parameter shall either be unspecified (i.e. contain a value of ‘null’), or shall specify the latest time at which the leading edge of the first bit of the CLTU may begin to be radiated.

3.6.2.7.3 If the latest-~~production~~radiation-time parameter is unspecified, the provider shall process the CLTU as long as the service instance is in state 3 (‘active’) and the production process is not halted.

3.6.2.7.4 If latest-radiation-time is specified, i.e., it is not ‘null’, the provider shall defer processing of a CLTU if the current production-status value is ‘interrupted’. Processing shall be deferred until either recovery from a temporary problem is accomplished, i.e., the production-status value changes to ‘operational’ before latest-radiation-time expires, or latest-radiation-time expires, in which case the provider shall discard the CLTU.

3.6.2.7.5 If latest-radiation-time equals earliest-radiation-time, radiation shall occur at this time.

3.6.2.7.6 If radiation has not begun at or before latest-radiation-time, an ‘sldu expired’ exception shall be notified to the user, and the CLTU shall not be radiated.

3.6.2.8 delay-time

3.6.2.8.1 The **delay-time** parameter shall contain the minimum radiation delay, in microseconds, between the CLTU transferred in this operation and the next CLTU.

3.6.2.8.2 The delay shall be measured from the trailing edge of the last radiated bit of this CLTU to the leading edge of the first bit of the next CLTU.

3.6.2.8.3 If `delay-time` contains a value of zero, radiation of the next CLTU may occur immediately.

3.6.2.8.4 The effect of `delay-time` varies depending on the setting of the `plop-in-effect` parameter.

a) ~~w~~When the PLOP in effect is PLOP-1 (reference [2])~~,~~:

   1) ~~c~~Command modulation shall be dropped between CLTUs for the period of time specified by `delay-time`~~,~~.

   2) ~~t~~The acquisition sequence and, if so configured, an idle sequence shall be transmitted prior to radiation of the next CLTU.

   3) ~~In deviation from reference [2] t~~The length of the acquisition sequence ~~can be modified by a managed parameter~~shall be determined by the value of the managed parameter `acquisition-sequence-length` (see table 3-1).

   NOTE – With a length set to zero the actually required acquisition sequence may be inserted into the CLTU proper enabling commanding of spacecraft, which do not fully adhere to PLOP-1 (as per reference [2]).

   4) The length of the idle sequence shall be determined by the value of the managed parameter `plop-1-idle-sequence-length` (see table 3-1).

   5) If so configured, after radiation of the CLTU, an idle sequence shall be radiated.

   ~~time for radiation of the acquisition sequence shall be in addition to the time specified by `delay-time`;~~

   6) The time between the radiation of two consecutive CLTUs shall be the time for radiation of the acquisition sequence added to the time specified by the `delay-time` parameter.

b) ~~w~~When the PLOP in effect is PLOP-2 (reference [2])~~,~~:

   ~~1) an idle sequence shall be radiated between CLTUs, for the period of time specified by `delay-time`,~~

   ~~2) the time specified by `delay-time` shall be increased to an integral number of bit times.~~

   1) After completion of the uplink sweep the acquisition sequence shall be transmitted prior to the radiation of the first CLTU or first idle sequence.

   2) The length of the acquisition sequence shall be determined by the value of the managed parameter `acquisition-sequence-length` (see table 3-1).

3) An idle sequence shall be radiated between CLTUs, for at least the period of time specified by the `delay-time` parameter.

3.6.2.9 report

The **`report`** parameter shall specify whether the provider shall invoke the CLTU-ASYNC-NOTIFY operation upon completion of the radiation of the CLTU, and shall contain one of the following values:

a) 'produce report'—invoke a CLTU-ASYNC-NOTIFY operation upon completion of the radiation;

b) 'do not produce report'—do not invoke the CLTU-ASYNC-NOTIFY operation.

3.6.2.10 data

The **`data`** parameter shall contain a CLTU for radiation to the space element of a mission.

3.6.2.11 cltu-buffer-available

The **`cltu-buffer-available`** parameter shall specify the remaining number of octets available for buffering CLTUs.

NOTE – Real-world implementations in general will have a small additional buffer at the modulator input in order to prevent data underflow in particular at high command rates and when the interface towards the modulator is asynchronous, e.g., based on TCP. Users should be aware that such additional buffer is not reflected in the reported remaining number of octets available for buffering CLTUs, as such modulator buffer is part of service production and outside the service provisioning.

3.6.2.12 result

The **`result`** parameter shall specify the result of the CLTU-TRANSFER-DATA operation and shall contain one of the following values:

a) 'positive result'—the CLTU-TRANSFER-DATA operation has been performed by the provider and the data will be buffered until processing is due;

b) 'negative result'—the CLTU-TRANSFER-DATA operation has not been performed by the provider, for the reason returned in the `diagnostic` parameter.

3.6.2.13 diagnostic

3.6.2.13.1 If result is 'negative result', **diagnostic** shall be present in the return and its value shall be one of the following:

a) 'duplicate invoke-ID' —the value of the invoke-ID parameter is the same as the invoke-ID value of a previous, outstanding operation;

b) 'unable to process'—the provider cannot process CLTUs for one of the following reasons:

   1) the provider has been taken out of service for an indefinite period by management action, i.e., production-status is 'halted';

   2) the production-status is 'interrupted', and the provider has reported the fault condition to the user via a CLTU-ASYNC-NOTIFY operation containing a notification-type of 'production interrupted';

NOTE – The provider may defer notifying the user when the production-status becomes 'interrupted', until a CLTU becomes ready to radiate. In this case, the provider continues to accept CLTU-TRANSFER-DATA operations so long as the notification has not been sent. See annex G.

   3) the production-status is 'operational' but the provider is blocked because of an earlier fault: the provider has reported the fault condition to the user by a CLTU-ASYNC-NOTIFY operation containing the notification-type value 'production interrupted' and has reported the recovery from the fault condition to the user by a CLTU-ASYNC-NOTIFY operation containing the notification-type value 'production operational';

   4) the production-status is 'operational', but the provider is blocked due to an expired CLTU: the provider has reported the fault condition to the user via a CLTU-ASYNC-NOTIFY operation containing a notification-type of 'sldu expired';

   5) the production-status is 'operational', but the provider has temporarily suspended production to process a CLTU-THROW-EVENT invocation: no fault condition has been reported to the user;

c) 'unable to store'—there is not enough buffer space available to store this CLTU;

d) 'out of sequence'—the value of the cltu-identification parameter is not equal to the value expected by the provider, which is one of the following:

   1) in the case of the first CLTU-TRANSFER-DATA operation following a CLTU-START, the value specified by the user in the first-cltu-identification parameter of the CLTU-START operation;

2) otherwise, the value of the `cltu-identification` parameter specified by the provider in the last CLTU-TRANSFER-DATA return;

e) 'inconsistent time range'— the time specified in the `earliest-radiation-time` parameter is later than the time specified in the `latest-radiation-time` parameter;

f) 'invalid time'—the production time window is invalid, for one of the following reasons:

1) the period from `earliest-`~~`production`~~`radiation-time` to `latest-`~~`production`~~`radiation-time` does not overlap with the range of times for which service production is scheduled;

2) the period from `earliest-`~~`production`~~`radiation-time` to `latest-`~~`production`~~`radiation-time` does not overlap with the service instance provision period;

NOTE – The production may be scheduled to terminate earlier than the service instance provision period ends. An SLE Complex may do so to have the production engine available for support of a different mission as soon as possible, but permitting the users of the previous production period some extra time to retrieve for example a status report reflecting the final accounting information.

g) 'late sldu'—latest-radiation-time is earlier than the time the CLTU-TRANSFER-DATA operation is received by the provider;

h) 'invalid delay time'—the value of the delay-time is ~~outside the range set in the service package~~less than the `minimum-delay-time` value set by Service Management;

i) 'CLTU error'—the provider has performed error checks as provided in the service agreement and has determined that this CLTU is in error; for example the CLTU exceeds the maximum size allowed for this service instance;

j) 'other reason'— the reason for the negative result will have to be found by other means.

3.6.2.13.2 If `result` is 'positive result', the `diagnostic` parameter shall not be present in the return.

3.6.3 EFFECTS

3.6.3.1 If `result` is 'positive result', the CLTU-TRANSFER-DATA operation shall have the following effects:

a) the provider shall buffer the CLTU until it is due to be processed;

3.7 CLTU-ASYNC-NOTIFY

3.7.1 PURPOSE

3.7.1.1 The CLTU service provider shall invoke the CLTU-ASYNC-NOTIFY operation to notify the user of an event affecting the production of the Forward CLTU service.

NOTE – Notification of events may be of value to the user in understanding specific provider behavior, such as an interruption of the command radiation.

3.7.1.2 CLTU-ASYNC-NOTIFY shall be an unconfirmed operation.

NOTE – Notifications from the provider are delivered to the user asynchronously to the flow of CLTU-TRANSFER-DATA operations from the user to the provider.

3.7.1.3 The CLTU-ASYNC-NOTIFY operation is valid in states 2 (‘ready’) and 3 (‘active’) and shall be invoked only by the provider.

3.7.2 INVOCATION~~, RETURN, AND~~ AND PARAMETERS

3.7.2.1 General

The parameters of the CLTU-ASYNC-NOTIFY operation shall be present in the invocation as specified in table 3-7.

Table 3-7: CLTU-ASYNC-NOTIFY Parameters

Parameters	Invocation
invoker-credentials	M
notification-type	M
event-thrown-identification	C
cltu-last-processed	M
cltu-last-OK	M
cltu-status	C
radiation-start-time	C
radiation-stop-time	C
production-status	M
uplink-status	M

3.7.2.2 invoker-credentials

The **invoker-credentials** parameter shall provide information that enables the performer to authenticate the CLTU-ASYNC-NOTIFY invocation (see 3.1.5).

3.7.2.3 notification-type

The **notification-type** parameter shall describe the event being notified to the user and shall contain one of the following values:

a) 'cltu radiated'—the CLTU identified by the value of the cltu-last-processed parameter successfully completed radiation. This value of notification-type shall be used only if the value of report in the associated CLTU-TRANSFER-DATA invocation was 'produce report'.

b) 'sldu expired'—radiation of the CLTU identified by the value of the cltu-last-processed parameter did not begin by the time specified in the latest-radiation-time parameter of the associated CLTU-TRANSFER-DATA invocation. No further CLTUs shall be radiated; buffered CLTUs shall be discarded; and further CLTU-TRANSFER-DATA invocations shall be blocked, i.e., rejected with an 'unable to process' diagnostic.

NOTE – After the 'sldu expired' event, the service user has to clear the blocking by invoking a CLTU-STOP operation. In order to resume the transfer and radiation of CLTUs, the user must successfully invoke CLTU-START.

c) 'production interrupted'—the production process has stopped due to a condition that may be temporary. This event occurs and the notification shall be sent

1) for notification mode 'immediate' when the production-status changes to 'interrupted'; or

2) for notification mode 'deferred' when production status is interrupted and either a CLTU is in status 'radiation started' or the radiation of the CLTU has to be started based on the specified earliest-radiation-time and latest-radiation-time parameters.

No further CLTUs shall be radiated; buffered CLTUs shall be discarded; and, in state 3 ('active'), further CLTU-TRANSFER-DATA invocations shall be blocked, i.e., rejected with an 'unable to process' diagnostic.

NOTE – ~~The production process achieves operational status after initial establishment blocking~~The user can unblock the service instance by invoking a CLTU-STOP operation. After the condition causing the 'production interrupted' event is corrected, the provider notifies the user by means of a 'production operational' notification. The user can resume the transfer and radiation of CLTUs after successfully invoking CLTU-START.

d) 'production halted'—the production process has been stopped by management action. No further CLTUs shall be radiated; buffered CLTUs shall be discarded; and, in state 3 ('active'), further CLTU-TRANSFER-DATA invocations shall be blocked, i.e., rejected with an 'unable to process' diagnostic.

NOTES

3.7.2.11.2 The `uplink-status` parameter shall contain one of the following values:

~~NOTE – Due to the loss of the uplink as notified by a) to c) below it is most likely no longer possible to uplink a CLTU to the spacecraft. Nonetheless the production status of the Forward CLTU service remains 'operational'. Recovery from this condition will require user/provider interaction outside the scope of this Recommended Standard.~~

a) 'uplink status not available'—no CLCWs from the spacecraft have been received by the provider;

b) 'no rf available'—the provider has received at least one CLCW; in the last CLCW received by the provider, the bit that flags 'No RF Available' was set to '1';

c) 'no bit lock'—the provider has received at least one CLCW; in the last CLCW received by the provider, the bit that flags 'No RF Available' was set to '0', and the bit that flags 'No Bit Lock' was set to '1';

d) 'nominal'—the provider has received at least one CLCW; in the last CLCW received by the provider, the bit that flags 'No RF Available' was set to '0', and the bit that flags 'No Bit Lock' was set to '0'.

3.7.3 EFFECTS

3.7.3.1 If `notification-type` indicates 'sldu expired', 'production interrupted', or 'production halted':

a) no further CLTUs shall be radiated;

b) buffered CLTUs shall be discarded; and

c) further CLTU-TRANSFER-DATA invocations shall be rejected.

3.7.3.2 There shall be no effect for any other values of `notification-type`.

3.7.3.3 The provider shall remain in its original state, i.e., state 2 ('ready') or state 3 ('active').

a) ‘positive result’—the CLTU-GET-PARAMETER operation has been performed, and the value of the specified CLTU service parameter is provided in the return to the user;

b) ‘negative result’—the CLTU-GET-PARAMETER operation has not been performed for the reason specified in the diagnostic parameter.

Table 3-11: Forward CLTU Parameters

Parameter	Description
bit-lock-required	If the value is ‘yes’, the ‘No bit lock’ flag in the CLCW must be false in order for the provider to set ~~the production status~~production-status to ‘operational’.
delivery-mode	‘fwd online’
expected-cltu-identification	The expected value of the cltu-identification parameter to be received in the next CLTU-TRANSFER-DATA invocation. If no CLTU-START has been received, zero shall be returned as the default value of this parameter.
expected-event-invocation-identification	The expected value of the event-invocation-identification parameter to be received in the next CLTU-THROW-EVENT invocation. The initial value of this parameter is zero.
maximum-cltu-length	The size, in octets, of the maximum-length CLTU that will be accepted by the provider for this service instance.
modulation-frequency	The frequency of the primary modulation of the RF carrier, expressed in tenths of Hertz. Primary modulation means the subcarrier frequency, when applicable, otherwise the frequency of direct data modulation.
modulation-index	The angle by which the RF carrier is phase shifted with respect to the un-modulated RF carrier, expressed in milli-radians (10^{-3} rad).
plop-in-effect	The physical layer operation procedure (PLOP) being used: ‘PLOP-1’ or ‘PLOP-2’.
reporting-cycle	The current setting of the reporting cycle for status reports (see 3.8 and 3.9): the value is ‘null’ if cyclic reporting is off, otherwise it is the time (in seconds) between successive CLTU-STATUS-REPORT invocations (see ~~3.8.2.6~~3.8).
return-timeout-period	~~M~~The maximum time period (in seconds) permitted ~~between the invocation of a~~from when a confirmed Forward CLTU operation ~~and the receipt by the invoker of the return (for confirmed operations).~~is invoked until the return is received by the invoker (~~S~~see 4.1.3).

rf-available-required	If the value is 'yes', the 'No RF available' flag in the CLCW must be false in order for the provider to set ~~the production status~~production-status to 'operational'.
subcarrier-to-bit-rate-ratio	When subcarrier modulation is used, the value represents the ratio of the subcarrier frequency to the uplink data rate (i.e., the bit rate). A value of one indicates that data will be directly modulated onto the carrier.

3.10.2.8 diagnostic

3.10.2.8.1 If result is 'negative result', **diagnostic** shall be present and shall contain one of the following values:

a) 'duplicate invoke-ID'—the value of the invoke-ID parameter is the same as the invoke-ID value of a previous, outstanding operation;

b) 'unknown parameter'—the value of cltu-parameter does not identify an CLTU service parameter that is recognized by the service provider;

c) 'other reason'—the reason for the negative result will have to be found by other means.

3.10.2.8.2 If result is 'positive result', the diagnostic parameter shall not be present in the return.

3.10.3 EFFECTS

3.10.3.1 If result is 'positive result', the value of the CLTU service parameter specified by the cltu-parameter parameter shall be returned to the user in the return.

3.10.3.2 If result is 'negative result', no CLTU service parameter specified by the cltu-parameter parameter shall be returned to the user.

3.10.3.3 The provider shall remain in its original state, i.e., state 2 ('ready') or state 3 ('active').

4 CLTU PROTOCOL

4.1 GENERIC PROTOCOL CHARACTERISTICS

NOTE – This section specifies the handling of invalid SLE-PDUs and other failures affecting the protocol.

4.1.1 UNEXPECTED SLE PROTOCOL DATA UNIT

If the peer application entity sends an invocation or return not allowed in the current state of the performer, the performer shall abort the association by invoking the CLTU-PEER-ABORT operation with the `diagnostic` parameter set to 'protocol error'.

4.1.2 INVALID OR UNDECODABLE PROTOCOL DATA UNIT

If the application entity receives an invocation or return that contains an unrecognized operation type, contains a parameter of a wrong type, or is otherwise not decodable, the application entity shall abort the association by invoking the CLTU-PEER-ABORT operation with the `diagnostic` parameter set to 'encoding error'.

4.1.3 MISSING RETURN

For confirmed operations, if the invoker does not receive the return from the performer within the return-timeout-period specified by service management, the invoker shall abort the association by invoking the CLTU-PEER-ABORT operation with the `diagnostic` parameter set to 'return timeout'.

NOTES

1 The return-timeout-period shall be chosen taking into account performance of user and provider applications as well as the delays introduced by the underlying communications service.

2 In order to provide responsive service and short return-timeout-periods, the generation of the return from an operation must not depend on any human interaction.

3 After invoking the CLTU-UNBIND operation, the initiator must not invoke any further operations with the exception of the case addressed in 3.3.1.4 nor send any returns. The responder is not required to send any pending returns after having received the CLTU-UNBIND invocation. Therefore, following an CLTU-UNBIND invocation, the 'missing return' event may occur.

Table 4-1: Provider Behavior

	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)

1~~.~~	(cltuBindInvocation)	IF “positive result”
		THEN (+cltuBindReturn) → 2
		ELSE (-cltuBindReturn)
			~~IF “same service instance”~~
			~~THEN (-cltuBindReturn (‘already bound’))~~
			~~ELSE~~ {peer abort ~~(~~‘protocol error’~~)~~} → 1
				~~IF “same service instance”~~
				~~THEN (-cltuBindReturn (‘already bound’))~~
				~~ELSE~~ {peer abort ~~(~~‘protocol error’~~)~~} → 1

2~~.~~	‘end of service instance provision period’	{clean up}	{peer abort ~~(~~‘end of service instance provision period’~~)~~} → 1	{peer abort ~~(~~‘end of service instance provision period’~~)~~} → 1

3~~.~~	(cltuUnbindInvocation)	[ignore]	(cltuUnbindReturn) → 1
			stop reporting-cycle timer
			IF “end”
			THEN release resources
			ELSE [ignore]
				{peer abort ~~(~~‘protocol error’~~)~~} → 1

4~~.~~	(cltuStartInvocation)	[ignore]	IF “positive result”
			THEN (+cltuStartReturn) → 3
			ELSE (-cltuStartReturn)
				{peer abort ~~(~~‘protocol error’~~)~~} → 1

5~~.~~	(cltuStopInvocation)	[ignore]	{peer abort ~~(~~‘protocol error’~~)~~} → 1	IF “positive result”
				THEN {initiate stop} → 2
				ELSE (-cltuStopReturn)

6~~.~~	(cltuTransferDataInvocation)	[ignore]	{peer abort ~~(~~‘protocol error’~~)~~} → 1	IF “positive result” .AND. (.NOT. “service instance blocked”)
				THEN buffer CLTU
				(+cltuTransferDataReturn)
				ELSE discard CLTU
				(-cltuTransferDataReturn)

	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)

7~~.~~	(cltuScheduleStatusReportInvocation)	[ignore]	IF “positive result”
			THEN (+cltuScheduleStatusReportReturn)
			IF “immediately”
			THEN {immediate report}
			ELSE IF “periodically”
			THEN {periodic report}
			ELSE stop reporting-cycle timer
			ELSE (-cltuScheduleStatusReportReturn)
				IF “positive result”
				THEN (+cltuScheduleStatusReportReturn)
				IF “immediately”
				THEN {immediate report}
				ELSE IF “periodically”
				THEN {periodic report}
				ELSE stop reporting-cycle timer
				ELSE (-cltuScheduleStatusReportReturn)

8~~.~~	‘reporting-cycle timer expired’	Not applicable	{periodic report}	{periodic report}
9~~.~~	‘return-timeout-period timer <n> expired’	Not applicable	{peer abort ~~(~~‘return timeout’~~)~~} → 1	{peer abort ~~(~~‘return timeout’~~)~~} → 1

10~~.~~	(cltuGetParameterInvocation)	[ignore]	IF “positive result”
			THEN (+cltuGetParameterReturn)
			ELSE (-cltuGetParameterReturn)
				IF “positive result”
				THEN (+cltuGetParameterReturn)
				ELSE (-cltuGetParameterReturn)

11~~.~~	(cltuThrowEventInvocation)	[ignore]	IF “positive result”
			THEN (+cltuThrowEventReturn)
			forward event to Complex Management
			ELSE (-cltuThrowEventReturn)
				IF “positive result”
				THEN (+cltuThrowEventReturn)
				forward event to Complex Management
				ELSE (-cltuThrowEventReturn)

12~~.~~	‘cltu radiated’	[ignore]	IF “report”
			THEN {notify ~~(~~‘cltu radiated’~~)~~}
			ELSE [ignore]
				IF “report”
				THEN {notify ~~(~~‘cltu radiated’~~)~~}
				ELSE [ignore]

13~~.~~	‘sldu expired’	IF “continue”
		THEN clear CLTU buffer
		ELSE [ignore]
			Not applicable	{notify ~~(~~‘sldu expired’~~)~~ and block}

14~~.~~	‘production interrupted’	IF “continue”
		THEN clear CLTU buffer
		ELSE [ignore]
			{notify ~~(~~‘production interrupted’~~)~~ and clear}
			set “notify production operational” to TRUE
				{notify ~~(~~‘production interrupted’~~)~~ and block}
				set “notify production operational” to TRUE

15~~.~~	‘production halted’	IF “continue”
		THEN clear CLTU buffer
		ELSE [ignore]
			{notify ~~(~~‘production halted’~~)~~ and clear}
			set “notify production operational” to TRUE
				{notify ~~(~~‘production halted’~~) ~~ and block}
				set “notify production operational” to TRUE

16~~.~~	‘production operational’	[ignore]	IF “notify production operational”
			THEN {notify ~~(~~‘production operational’~~)~~}
			set “notify production operational” to FALSE
				IF “notify production operational”
				THEN {notify ~~(~~‘production operational’~~)~~}
				set “notify production operational” to FALSE

	Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
17~~.~~	‘buffer empty’	[ignore]	Not applicable	{notify ~~(~~‘buffer empty’~~)~~}
18~~.~~	‘action list completed’	Not applicable	{notify ~~(~~‘action list completed’~~)~~}	{notify ~~(~~‘action list completed’~~)~~}
19~~.~~	‘action list not completed’	Not applicable	{notify ~~(~~‘action list not completed’~~)~~}	{notify ~~(~~‘action list not completed’~~)~~}
20~~.~~	‘event condition evaluated to false’	Not applicable	{notify ~~(~~‘event condition evaluated to false’~~)~~}	{notify ~~(~~‘event condition evaluated to false’~~)~~}
21~~.~~	(cltuPeerAbortInvocation)	[ignore]	{clean up} → 1	{clean up} → 1

22~~.~~	‘protocol abort’	[ignore]	{clean up} → 1	IF “continue”
				THEN stop reporting-cycle timer → 1
				ELSE {clean up} → 1

23~~.~~	‘unsolicited invoke-ID’	[ignore]	{peer abort ~~(~~‘unsolicited invoke-ID’~~)~~} → 1	{peer abort ~~(~~‘unsolicited invoke-ID’~~)~~} → 1
24~~.~~	‘invalid SLE-PDU’	[ignore]	{peer abort ~~(~~‘encoding error’~~)~~} → 1	{peer abort ~~(~~‘encoding error’~~)~~} → 1
25~~.~~	‘unexpected SLE-PDU’	[ignore]	{peer abort ~~(~~‘protocol error’~~)~~} → 1	{peer abort ~~(~~‘protocol error’~~)~~} → 1
26~~.~~	‘not authenticated SLE-PDU’	[ignore]	[ignore]	[ignore]

Table 4-2: Event Description References

Event	Reference
‘action list completed’	3.7.2.3
‘action list not completed’	3.7.2.3
‘buffer empty’	3.7.2.3
‘cltu radiated’	3.7.2.3
‘end of service instance provision period’	3.12.2.2
‘event condition evaluated to false’	3.7.2.3
‘invalid SLE-PDU’	4.1.2
‘not authenticated SLE-PDU’	4.1.7
‘production halted’	3.7.2.3
‘production interrupted’	3.7.2.3
‘production operational’	3.7.2.3
‘protocol abort’	4.1.5.4
‘reporting-cycle timer expired’	3.8.2.6
‘return-timeout-period timer <n> expired’	4.1.3
‘sldu expired’	3.7.2.3
‘unexpected SLE-PDU’	4.1.1
‘unsolicited invoke-ID’	3.12.2.2

Table 4-3: Predicate Definitions

Predicate	Evaluates to TRUE if
“continue”	the `protocol-abort-mode` parameter value is ‘continue’
“end”	all checks on the (cltuUnbindInvocation) PDU are passed and the `unbind-reason` parameter value is ‘end’
“immediately”	all parameter checks on the (cltuScheduleStatusReportInvocation) PDU are passed and the `report-request-type` parameter value is ‘immediately’
“periodically”	all parameter checks on the (cltuScheduleStatusReportInvocation) PDU are passed and the `report-request-type` parameter value is ‘periodically’
“positive result”	all checks on the given invocation PDU are passed
“report”	the `report` parameter value in the associated (cltuTransferDataInvocation) PDU is ‘produce report’
~~“same service instance”~~	~~the `service-instance-identifier` value in the (cltuBindInvocation) PDU denotes a service instance that is already bound~~

Table 4-4: Boolean Flags

Flag Name	Initial Value
“service instance blocked”	FALSE
“notify production operational”	FALSE

Table 4-5: Compound Action Definitions

Name	Actions Performed
{clean up}	a) stop reporting-cycle timer b) reset reporting-cycle c) clear CLTU buffer d) set “service instance blocked” to FALSE
{immediate report}	a) (cltuStatusReportInvocation) b) stop reporting-cycle timer
{initiate stop}	a) clear CLTU buffer b) set “service instance blocked” to FALSE c) (+cltuStopReturn)
{notify ~~(~~‘xxxx’~~)~~}	(cltuAsyncNotifyInvocation) with `notification-type` set to ‘xxxx’; i.e., to the value corresponding to the incoming event
{notify ~~(~~‘xxxx’~~)~~ and clear}	a) (cltuAsyncNotifyInvocation) with `notification-type` set to ‘xxxx’; i.e., to the value corresponding to the incoming event b) clear CLTU buffer
{notify ~~(~~‘xxxx’~~)~~ and block}	a) (cltuAsyncNotifyInvocation) with `notification-type` set to ‘xxxx’; i.e., to the value corresponding to the incoming event b) set “service instance blocked” to TRUE c) clear CLTU buffer
{peer abort ~~(~~‘xxxx’~~)~~}	a) (cltuPeerAbortInvocation) with `diagnostic` set to ‘xxxx’ b) stop reporting-cycle timer c) reset reporting-cycle d) clear CLTU buffer e) set “service instance blocked” to FALSE
{periodic report}	a) (cltuStatusReportInvocation) b) set reporting-cycle timer to the `reporting-cycle` value in the most recent (cltuScheduleStatusReportInvocation) c) start reporting-cycle timer

```
ParameterName                       ::=   INTEGER
{  apidList                               (2)
,  bitLockRequired                        (3)
,  blockingTimeoutPeriod                  (0)
,  blockingUsage                          (1)
,  bufferSize                             (4)
,  deliveryMode                           (6)
,  directiveInvocation                    (7)
,  directiveInvocationOnline              (108)
,  expectedDirectiveIdentification        (8)
,  expectedEventInvocationIdentification  (9)
,  expectedSlduIdentification             (10)
,  fopSlidingWindow                       (11)
,  fopState                               (12)
,  latencyLimit                           (15)
,  mapList                                (16)
,  mapMuxControl                          (17)
,  mapMuxScheme                           (18)
,  maximumFrameLength                     (19)
,  maximumPacketLength                    (20)
,  maximumSlduLength                      (21)
,  modulationFrequency                    (22)
,  modulationIndex                        (23)
,  permittedControlWordTypeSet            (101)
,  permittedGvcidSet                      (24)
,  permittedTcVcidSet                     (102)
,  permittedTransmissionMode              (107)
,  permittedUpdateModeSet                 (103)
,  plopInEffect                           (25)
,  reportingCycle                         (26)
,  requestedControlWordType               (104)
,  requestedFrameQuality                  (27)
,  requestedGvcid                         (28)
,  requestedTcVcid                        (105)
,  requestedUpdateMode                    (106)
,  returnTimeoutPeriod                    (29)
,  rfAvailable                            (30)
,  rfAvailableRequired                    (31)
,  segmentHeader                          (32)
,  subcarrierToBitRateRatio               (34)
,  timeoutType                            (35)
,  timerInitial                           (36)
,  transmissionLimit                      (37)
,  transmitterFrameSequenceNumber         (38)
,  vcMuxControl                           (39)
,  vcMuxScheme                            (40)
,  virtualChannel                         (41)
}

SlduStatusNotification  ::=   INTEGER
{  produceNotification           (0)
,  doNotProduceNotification      (1)
}

SpaceLinkDataUnit       ::=   OCTET STRING (SIZE (~~4~~1 .. 65536))
```

```
Time                     ::=   CHOICE
{  ccsdsFormat        [0]   TimeCCSDS
,  picoFormat         [1]   TimeCCSDSpico
}

TimeCCSDS                ::=   OCTET STRING (SIZE(8))
-- P-field is implicit (not present, defaulted to 41 hex
-- T-field:
-- 2 octets: number of days since 1958/01/01 00:00:00
-- 4 octets: number of milliseconds of the day
-- 2 octets: number of microseconds of the millisecond
--        (set to 0 if not used)
-- This definition reflects exactly the format of the CCSDS defined
-- time tag as used in spacelink data units (see Time Code Formats.
-- Recommendation for Space Data System Standards, CCSDS 301.0-B-3.
-- Blue Book.  Issue 3.  Washington, D.C.: CCSDS, January 2002).

TimeCCSDSpico            ::=   OCTET STRING (SIZE(10))
-- P-field is implicit (not present, defaulted to 42 hex
-- T-field:
-- 2 octets: number of days since 1958/01/01 00:00:00
-- 4 octets: number of milliseconds of the day
-- 4 octets: number of picoseconds of the millisecond
--        (set to 0 if not used)
-- This definition reflects exactly the format of the CCSDS defined
-- time tag as used in spacelink data units (see Time Code Formats.
-- Recommendation for Space Data System Standards, CCSDS 301.0-B-3.
-- Blue Book.  Issue 3.  Washington, D.C.: CCSDS, January 2002).

END
```

```
}

CltuIdentification               ::=   IntUnsignedLong

CltuLastOk                             ::=   CHOICE
{ noCltuOk                                   [0]      NULL
,  cltuOk                                    [1]      SEQUENCE
{  cltuIdentification               CltuIdentification
,  ~~radiationStopTime~~stopProductionTime                         Time
}
}

CltuLastProcessed                ::=   CHOICE
{ noCltuProcessed                      [0]   NULL
,  cltuProcessed                       [1]   SEQUENCE
{  cltuIdentification               CltuIdentification
,  ~~radiationStartTime~~startProductionTime                     ConditionalTime
,  cltuStatus                             CltuStatus
}
}

CltuNotification                    ::=   CHOICE
{  cltuRadiated                        [0]   NULL
,  slduExpired                            [1]   NULL
,  productionInterrupted                  [2]   NULL
,  productionHalted                    [3]   NULL
,  productionOperational                  [4]   NULL
,  bufferEmpty                            [5]   NULL
,  actionListCompleted                    [6]   EventInvocationId
,  actionListNotCompleted                 [7]   EventInvocationId
,  eventConditionEvFalse                  [8]   EventInvocationId
}


CltuParameterName                ::=   ParameterName
(  bitLockRequired
|  deliveryMode
|  expectedSlduIdentification
|  expectedEventInvocationIdentification
|  maximumSlduLength
|  modulationFrequency
|  modulationIndex
|  plopInEffect
|  reportingCycle
|  returnTimeoutPeriod
|  rfAvailableRequired
|  subcarrierToBitRateRatio
)

CltuStatus                             ::=   ForwardDuStatus
(  radiated
|  expired
|  interrupted
|  productionStarted
|  productionNotStarted
)

CurrentReportingCycle            ::=   CHOICE
{  periodicReportingOff          [0]   NULL
,  periodicReportingOn           [1]   ReportingCycle
```

```
{  operational                            (0)
,  configured                             (1)
,  interrupted                            (2)
,  halted                                 (3)
}

-- Divisor of the subcarrier frequency
-- If direct carrier modulation, the value is 1
SubcarrierDivisor                 ::=   IntPosShort

-- measured in seconds
TimeoutPeriod                           ::=   INTEGER (1 .. 600)

UplinkStatus                            ::=   INTEGER
{  uplinkStatusNotAvailable               (0)
,  noRfAvailable                          (1)
,  noBitLock                              (2)
,  nominal                                (3)
}

END
```

~~NOTE – The choice with the tag [9] from CltuNotification was removed. For any implementation that was built in accordance with the normative state table, this will not create an interoperability problem, since the choice with tag [9] got never invoked. Hence, neither on the encoding nor on the decoding size there will be a problem, even if an implementation was built using the old ASN.1 specification.~~

```
,  cltuStopReturn                   [3]   SleAcknowledgement
,  cltuScheduleStatusReportReturn         [5]
   SleScheduleStatusReportReturn
,  cltuGetParameterReturn                 [7]   CltuGetParameterReturn
,  cltuThrowEventReturn                   [9]   CltuThrowEventReturn
,  cltuTransferDataReturn                 [11]  CltuTransferDataReturn
,  cltuAsyncNotifyInvocation              [12]  CltuAsyncNotifyInvocation
,  cltuStatusReportInvocation             [13]  CltuStatusReportInvocation
,  cltuPeerAbortInvocation                [104] SlePeerAbort

}

-- =================================================================
-- The second part of the module definition contains the types
-- used by the CLTU-PDUs declared in the first part.
-- =================================================================

CltuAsyncNotifyInvocation     ::=   SEQUENCE
{  invokerCredentials               Credentials
,  cltuNotification                 CltuNotification
,  cltuLastProcessed                CltuLastProcessed
,  cltuLastOk                       CltuLastOk
,  productionStatus                 ProductionStatus
,  uplinkStatus                     UplinkStatus
}

CltuGetParameterReturn        ::=   SEQUENCE
{  performerCredentials                Credentials
,  invokeId                            InvokeId
,  result                              CHOICE
   {  positiveResult                   [0]   CltuGetParameter
   ,  negativeResult                   [1]   DiagnosticCltuGetParameter
   }
}

CltuStartReturn            ::=   SEQUENCE
{  performerCredentials                Credentials
,  invokeId                            InvokeId
,  result                              CHOICE
{  positiveResult                      [0]   SEQUENCE
      {  ~~radiationStartTime~~startProcuctionTime        Time
      ,  ~~radiationStopTime~~stopProductionTime    ConditionalTime
      }
   ,  negativeResult                   [1]   DiagnosticCltuStart
   }
}

CltuStatusReportInvocation    ::=   SEQUENCE
{  invokerCredentials                  Credentials
,  cltuLastProcessed                   CltuLastProcessed
,  cltuLastOk                          CltuLastOk
,  cltuProductionStatus                ProductionStatus
,  uplinkStatus                        UplinkStatus
,  numberOfCltusReceived               NumberOfCltusReceived
,  numberOfCltusProcessed              NumberOfCltusProcessed
,  numberOfCltusRadiated               NumberOfCltusRadiated
,  cltuBufferAvailable                 BufferSize
}

CltuThrowEventReturn          ::=   SEQUENCE
```

Other Sections	Optional / Mandatory
Provider Buffering Requirements (see 3.1.10)	Mandatory minimum size is 1024 maximum sized CLTUs where the maximum sized CLTU shall have at least a length of 4096 octets.
State transition table (see table 4-1)	Mandatory.
ASN.1 Protocol Specification (annex A)	All data types are mandatory. Implementations may vary as described in the annex.

Term	**Reference**
online delivery mode	reference [1]
Open System Interconnection (OSI)	reference [7]
operation	reference [1]
parameter	subsection 1.6.1.6.6
performance	subsection 1.6.1.6.7
performer	reference [1]
physical channel	reference [1]
Physical Layer Operations Procedure (PLOP)	reference [2]
port identifier	subsection 1.6.1.6.8
real system	reference [7]
responder	subsection 1.6.1.6.9
return	subsection 1.6.1.6.10
Service Access Point (SAP)	reference [7]
service instance provision period	subsection 1.6.1.6.11
service provider (provider)	reference [1]
service user (user)	reference [1]
SLE Complex	reference [1]
SLE Complex Management	reference [1]
SLE data channel	reference [1]
SLE functional group (SLE-FG)	reference [1]
SLE protocol data unit (SLE-PDU)	reference [1]
SLE service data unit (SLE-SDU)	reference [1]
SLE service package	reference [1]
~~SLE System~~	~~reference [1]~~
SLE transfer service instance	reference [1]
SLE transfer service production	reference [1]
SLE transfer service provision	reference [1]
SLE Utilization Management	reference [1]
space link	reference [1]
space link data channel	reference [1]
space link data unit (SL-DU)	reference [1]
space link session	reference [1]
unconfirmed operation	subsection 1.6.1.6.12

ANNEX E

INFORMATIVE REFERENCES

(INFORMATIVE)

[E1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-9. Yellow Book. Issue 9. Washington, D.C.: CCSDS, November 2003.

[E2] *Telecommand Summary of Concept and Rationale*. Report Concerning Space Data System Standards, CCSDS 200.0-G-6. Green Book. Issue 6. Washington, D.C.: CCSDS, January 1987.

[E3] *Cross Support Concept — Part 1: Space Link Extension Services*. Report Concerning Space Data System Standards, CCSDS 910.3-G-3. Green Book. Issue 3. Washington, D.C.: CCSDS, March 2006.

[E4] *Telecommand Part 1—Channel Service*. Recommendation for Space Data System Standards, CCSDS 201.0-B-3-S. Historical Recommendation. Issue 3-S. Washington, D.C.: CCSDS, (June 2000) August 2005.

[E5] *Telecommand Part 2—Data Routing Service*. Recommendation for Space Data System Standards, CCSDS 202.0-B-3-S. Historical Recommendation. Issue 3-S. Washington, D.C.: CCSDS, (June 2001) August 2005.

[E6] *Telecommand Part 2.1—Command Operation Procedures*. Recommendation for Space Data System Standards, CCSDS 202.1-B-2-S. Historical Recommendation. Issue 2-S. Washington, D.C.: CCSDS, (June 2001) August 2005.

[E7] *Telecommand Part 3—Data Management Service*. Recommendation for Space Data System Standards, CCSDS 203.0-B-2-S. Historical Recommendation. Issue 2-S. Washington, D.C.: CCSDS, (June 2001) August 2005.

[E8] *Space Link Extension—Internet Protocol for Transfer Services*. Recommendation for Space Data System Standards, CCSDS 913.1-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2008.

Table G-1: Production Status Changes and Notifications

Start Status	End Status	Cause of Status Change	Notification
Configured	Operational	Management action to make the production process operational; typically includes: completion of Uplink Sweep, and radiation of the Acquisition Sequence. Optionally, bit-lock and/or rf-~~lock~~available may be required before `production-status` can change to 'operational'. See table 3-11 and NOTE 1. If any error occurs that prevents the transition to production status 'operational' the production status reported will still stay 'configured'. If the status remains 'configured' for an extended period the user will have to check for the reasons by means outside the scope of this document.	'production operational'
Operational	Interrupted	Occurrence of a production fault detected by the provider.	'production interrupted'; but see NOTE 2 and 3
Interrupted	Operational	Maintenance action typically is required to correct a production fault. The CLTU `production-status` returns to 'operational' when the provider detects that the fault is corrected.	'production operational'
[Any]	Halted	Direct management action is required, such as an operator directive causing the provider to halt production.	'production halted'
Halted	Configured	Direct management action is required, such as an operator directive restoring the desired configuration and setting the `production-status` to 'configured'.	none

NOTES

1 The initial `production-status` value is 'configured'. When requested by the supported agency (either by schedule or via voice communication), the supporting agency starts CMM-1 by turning on the uplink and performing the nominal uplink sweep in accordance with the mission specific parameters. The `production-status` changes from 'configured' to 'operational':

a) at completion of the uplink sweep, if `plop-in-effect` is 'PLOP-1' and `rf-available-required` is set to 'no';

b) at completion of the uplink sweep, if plop-in-effect is ‘PLOP-1’, rf-available-required is set to ‘yes’ and the CLCW indicates ‘RF available’;

c) at completion of the radiation of the acquisition sequence, if plop-in-effect is ‘PLOP-2’, rf-available-required is set to ‘no’ and bit-lock-required is set to ‘no’;

d) at completion of the radiation of the acquisition sequence, if plop-in-effect is ‘PLOP-2’, rf-available-required is set to ‘yes’, the CLCW indicates ‘RF available’ and bit-lock-required is set to ‘no’;

e) at completion of the radiation of the acquisition sequence, if plop-in-effect is ‘PLOP-2’, rf-available-required is set to ‘no’, bit-lock-required is set to ‘yes’ and the CLCW flags ‘bit lock’;

f) at completion of the radiation of the acquisition sequence, if plop-in-effect is ‘PLOP-2’, rf-available-required is set to ‘yes’, bit-lock-required is set to ‘yes’ and the CLCW indicates both ‘RF available’ and ‘bit lock’.

2 The production-status changes to ‘interrupted’ when the checking of uplink-status is required (rf-available-required and/or bit-lock required set to ‘yes’) and uplink-status has no longer the required value.

3 This Recommended Standard does not require that a production interruption be notified immediately to the user (see 3.7.2.3). If a fault affecting production can be corrected before the provider attempts to process a CLTU, the ‘production interrupted’ notification need not be sent. It is an implementation choice whether always to send the ‘production interrupted’ notification immediately on occurrence of a production fault~~,~~ or to wait until a CLTU is ready to be radiated before sending the notification. The implementation choice is documented in the service ~~table~~configuration via the notification-mode parameter, which may have a value of ‘immediate’ or ‘deferred’, corresponding to immediate or deferred notification in event of a production interruption.

The effect of production-status on the processing of Forward CLTU transfer service invocations is specified in 3.1.10 through 3.12 and in table 4-1. The effects are summarized in table G-2 for convenient reference.

Table G-2: Effect of Production Status on Operations

Production Status	Operation	Effect	Diagnostic
Halted	CLTU-BIND	Rejected	‘out of service’
	CLTU-START	Rejected	‘out of service’
	CLTU-TRANSFER-DATA	Rejected	‘unable to process’
	Other operations	None specified	N/A
Configured	All operations	None specified	N/A
Operational	All operations	None specified	N/A
Interrupted	CLTU-START	Rejected	‘unable to comply’
	CLTU-TRANSFER-DATA	Rejected; but see NOTE	‘unable to process’
	Other operations	None specified	N/A

~~NOTE When the value of the notification-mode parameter is ‘deferred’, only after the provider notifies the user that a production interruption has occurred are CLTU-TRANSFER-DATA invocations rejected. See 3.7.2.3.~~

NOTE – When the value of the notification-mode parameter is ‘deferred’, CLTU-TRANSFER-DATA invocations are rejected only after the provider has notified the user that a production interruption has occurred. See 3.7.2.3.