



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Q.754

(06/97)

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN

Especificaciones del sistema de señalización N.º 7 –
Gestión del sistema de señalización N.º 7

**Definición de los elementos de servicio de
aplicación de gestión del sistema de
señalización N.º 7**

Recomendación UIT-T Q.754

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES DE LA SERIE Q DEL UIT-T
CONMUTACIÓN Y SEÑALIZACIÓN

SEÑALIZACIÓN EN EL SERVICIO MANUAL INTERNACIONAL	Q.1–Q.3
EXPLOTACIÓN INTERNACIONAL SEMIAUTOMÁTICA Y AUTOMÁTICA	Q.4–Q.59
FUNCIONES Y FLUJOS DE INFORMACIÓN PARA SERVICIOS DE LA RDSI	Q.60–Q.99
CLÁUSULAS APLICABLES A TODOS LOS SISTEMAS NORMALIZADOS DEL UIT-T	Q.100–Q.119
ESPECIFICACIONES DE LOS SISTEMAS DE SEÑALIZACIÓN N.º 4 Y N.º 5	Q.120–Q.249
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 6	Q.250–Q.309
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R1	Q.310–Q.399
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R2	Q.400–Q.499
CENTRALES DIGITALES	Q.500–Q.599
INTERFUNCIONAMIENTO DE LOS SISTEMAS DE SEÑALIZACIÓN	Q.600–Q.699
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 7	Q.700–Q.849
Generalidades	Q.700
Parte transferencia de mensajes	Q.701–Q.709
Parte control de la conexión de señalización	Q.711–Q.719
Parte usuario de telefonía	Q.720–Q.729
Servicios suplementarios de la RDSI	Q.730–Q.739
Parte usuario de datos	Q.740–Q.749
Gestión del sistema de señalización N.º 7	Q.750–Q.759
Parte usuario de la RDSI	Q.760–Q.769
Parte aplicación de capacidades de transacción	Q.770–Q.779
Especificaciones de las pruebas	Q.780–Q.799
Interfaz Q3	Q.800–Q.849
SISTEMA DE SEÑALIZACIÓN DIGITAL DE ABONADO N.º 1	Q.850–Q.999
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
RED INTELIGENTE	Q.1200–Q.1999
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999

Para más información, véase la Lista de Recomendaciones del UIT-T.

RECOMENDACIÓN UIT-T Q.754

DEFINICIÓN DE LOS ELEMENTOS DE SERVICIO DE APLICACIÓN DE GESTIÓN DEL SISTEMA DE SEÑALIZACIÓN N.º 7

Resumen

Esta Recomendación define el elemento de servicio de aplicación utilizado por las funciones de gestión MRVT, SRVT y CVT definidas en la Recomendación Q.753. El elemento de servicio de aplicación define la información de gestión utilizada por estas funciones en mensajes a través de la red del SS N.º 7.

El elemento de servicio se interconecta con capacidades de transacción para proporcionar los servicios utilizados por el usuario OMASE (definido en la Recomendación Q.753), a efectos de permitir la comunicación SS N.º 7 entre nodos mediante las pruebas MRVT, SRVT o CVT.

Las revisiones principales de la versión 1993 de la Recomendación son:

- a) revisión de las reglas de compatibilidad para permitir el transporte transparente de parámetros no reconocidos;
- b) definición de un nuevo parámetro de mensajes MRVT y SRVT para especificar cuál es la información requerida en todo mensaje MRVR o SRVR respectivo;
- c) definición de un nuevo mensaje MRVR y SRVR que se ha de utilizar si se requiere información fuera de la especificada para los mensajes MRVR y SRVR de la versión 1993;
- d) definición de nuevos parámetros de mensaje MRVT y MRVR para indicar rutas prioritarias;
- e) definición de nuevos parámetros MRVA, MRVR, SRVA y SRVR para permitir la devolución de parámetros no reconocidos en los mensajes MRVT o SRVT;
- f) definición de un nuevo parámetro MRVT que requiere nodos para comprobar si tienen una ruta hacia el iniciador de prueba a través del nodo por el que recibieron el mensaje MRVT (esto permite la verificación de rutas simétricas).

Orígenes

La Recomendación UIT-T Q.754, ha sido revisada por la Comisión de Estudio 11(1997-2000) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 5 de junio de 1997.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT ha recibido/no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 1997

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

ÍNDICE

Página

1	Introducción	1
2	MTP	2
2.1	Prueba de verificación del encaminamiento MTP (MRVT).....	2
2.1.1	testRoute Action (acción probar ruta).....	2
2.1.2	routeTrace Event (suceso de rastreo de ruta).....	5
2.1.3	routeTraceNew (nuevo rastreo de ruta).....	7
3	Parte control de la conexión de señalización (SCCP).....	9
3.1	ASE de la prueba de verificación del encaminamiento SCCP (SRVT).....	9
3.1.1	testRouteAction (acción probar ruta).....	10
3.1.2	routeTrace Event (suceso de rastreo de ruta).....	15
3.1.3	Suceso routeTraceNew (nuevo rastreo de ruta).....	17
4	Gestión de circuitos	21
4.1	Elemento de servicio de aplicación de la prueba de validación de circuitos (CVT)...	21
4.1.1	cktValidTest CnfAction (acción confirmada prueba de validación de circuito).....	21
4.1.2	Action Arguments (argumentos de acción).....	21
4.1.3	Action Results (resultados de acción).....	21
4.1.4	Specific Error (error específico)	22
5	Capacidades de transacción (TC).....	22
6	Definiciones generales	23
6.1	Objetos y operaciones	23
6.2	Primitivas y procedimientos del protocolo OMASE	23
6.2.1	Consideraciones generales.....	23
6.2.2	OM-EVENT-REPORT (OM-información de suceso)	23
6.2.3	OM-CONFIRMED-ACTION (OM-acción confirmada).....	25
6.3	Sintaxis abstracta del protocolo OMASE	30
	Anexo A – Utilización de las interfaces de las primitivas	41

Recomendación Q.754

DEFINICIÓN DE LOS ELEMENTOS DE SERVICIO DE APLICACIÓN DE GESTIÓN DEL SISTEMA DE SEÑALIZACIÓN N.º 7

(revisada en 1997)

1 Introducción

Debe señalarse que en caso de discrepancia entre las Recomendaciones Q.753 y Q.754, la segunda de ellas tendrá prioridad.

La presente Recomendación define los elementos de servicio de aplicación de la parte operaciones, mantenimiento y administración (OMASE, *operations, maintenance and administration part application service element*). La OMASE proporciona los servicios invocados utilizando las primitivas OM-EVENT-REPORT (OM-INFORME DE EVENTO) y OM-CONFIRMED-ACTION (OM-ACCIÓN-CONFIRMADA) a través de la frontera entre el usuario OMASE y la OMASE (véase en la Recomendación Q.753 el diagrama y la correspondencia entre los servicios invocados por el usuario OMASE y los servicios de OMASE).

Los servicios OMASE se derivan de los definidos en el protocolo común de información de gestión (CMIP, *common management information protocol*)¹.

Las primitivas OMASE se definen en la cláusula 6. La sintaxis formal indicada en la figura 3 utiliza los macros OPERATION y ERROR de capacidades de transacción (TC, *transaction capabilities*). En la cláusula 6 también se define el interfuncionamiento entre OMASE y TC.

OMASE proporciona las operaciones que permiten a la administración de la red llevar a cabo, a través del proceso de gestión OMAP y el usuario OMASE, la prueba de verificación del encaminamiento por la parte transferencia de mensajes (MRVT, *MTP routing verification tests*) y la prueba de verificación de encaminamiento por la parte control de conexión de señalización (SRVT, *SCCP routing verification tests*), y la prueba de validación de circuitos (CVT, *circuit validation tests*). La presente Recomendación contiene la definición de ASE para MRVT, SRVT y CVT.

La SRVT indicada se refiere a la prueba especificada en 3.2.2/Q.753.

Los argumentos utilizados en las primitivas a través de la frontera entre el proceso de gestión OMAP y el usuario OMASE y en las primitivas a través de la frontera entre el usuario OMASE y la OMASE y entre la OMASE y las TC contienen la misma información si tienen el mismo nombre. Dichos argumentos aparecen definidos en esta Recomendación.

Los mensajes entre puntos de señalización se codifican utilizando las reglas de codificación básicas (BER, *basic encoding rules*) ASN.1 y los parámetros de las cadenas de octetos se codifican como elementos de primitivas (no construidos).

¹ CMIP se define en ISO/CEI 9596 y en la Recomendación X.711.

2 MTP

2.1 Prueba de verificación del encaminamiento MTP (MRVT)²

La prueba MRVT iniciada en el origen de la prueba hace que se utilice una primitiva OM-CONFIRMED-ACTION del usuario OMASE a la OMASE, que incluye la instrucción testRoute como parámetro. Si se solicita un rastreo de las rutas o hay una avería, se invoca la primitiva OM-EVENT-REPORT en el originador de la prueba desde la OMASE, que incluye como parámetro el suceso routeTrace o el suceso routeTraceNew.

La instrucción testRoute se especifica utilizando el macro CNF-ACTION definido en la figura 3, y el suceso routeTrace se especifica utilizando el macro EVENT definido en la figura 3. La figura 3 define el módulo OMASE así como todos los parámetros utilizados en el protocolo.

Para MRVT, ObjectClass indica los cuadros de encaminamiento MTP y ObjectInstance contiene el código de punto del destino de la prueba. En testRoute Action se utiliza el mensaje BEGIN (MRVT), y el resultado (MRVA) se devuelve en END. routeTrace Event (MRVR) utiliza un mensaje BEGIN con fin previamente acordado.

2.1.1 testRoute Action (acción probar ruta)

testRoute Action se invoca para iniciar una prueba de verificación de encaminamiento por la MTP. En el nodo iniciador, esta invocación es solicitada por la Administración a través del usuario del sistema de información de gestión (MIS, *management information system*) o una interfaz local, mediante el proceso de gestión OMAP y el usuario OMASE. En los nodos siguientes, la acción es solicitada implícitamente al recibir una invocación testRoute Action. Una respuesta positiva indica que la prueba se ha completado con éxito en el punto en que se invocó e, implícitamente, en todos los puntos siguientes donde se invocó la prueba. Se devuelve una indicación de fallo para informar que la prueba ha fallado en este nodo o en otro siguiente.

testRoute CNF-ACTION	Temporizador = T1	Clase = 1	Código = 00000001
----------------------	-------------------	-----------	-------------------

Véase la figura 3.

2.1.1.1 testRoute Action Arguments (argumentos de acción probar ruta)

2.1.1.1.1 initiatingSP (punto de señalización iniciador)

initiatingSP identifica al solicitante original de la prueba. Es del tipo PointCode, definido como una cadena de octetos.

Parámetro	Código
initiatingSP	10000000.
Contenido	
El bit 0 contiene el primer bit del código de punto.	
El bit 1 contiene el segundo bit del código de punto, etc.	

² Para la descripción de la notación formal véanse las Recomendaciones desde la X.680 a la X.683 y la X.690 (véanse también las Recomendaciones X.208 y X.209).

2.1.1.1.2 traceRequested (rastreo solicitado)

traceRequested indica que debe informarse al originador sobre el rastreo de todas las rutas utilizadas para llegar al destino (el suceso routeTrace se describe en 2.1.2). Es del tipo BOOLEAN.

<i>Parámetro</i>	<i>Código</i>
traceRequested	10000001
<i>Contenido</i>	<i>Significado</i>
TRUE (= 1)	Se solicitó el rastreo, devolver información de rastreo sobre éxito/fracaso.
FALSE (= 0)	No se solicitó el rastreo, devolver información solamente si fracasa el rastreo.

2.1.1.1.3 threshold (umbral)

El originador fija un nivel de umbral máximo de puntos de señalización (SP, *signalling points*) que pueden cruzarse en el transcurso de la prueba [incluido el iniciador si es un punto de transferencia de señalización (STP)]. Esto ayuda a detectar rutas demasiado largas. Este umbral es un número entero de puntos de señalización, por lo que es del tipo INTEGER.

<i>Parámetro</i>	<i>Código</i>
threshold	10000010

2.1.1.1.4 pointCodesTraversed (códigos de punto atravesados)

Cuando se cruza cada punto de señalización intermedio, éste añade su propio código de punto a la lista de códigos de puntos atravesados. Esto ayuda a detectar bucles y es también una información útil en caso de un fallo o si se solicita el rastreo de una ruta. Es una lista de códigos de punto, por lo que es del tipo PointCodeList.

<i>Parámetro</i>	<i>Código</i>
pointCodesTraversed	10100011
<i>Contenido</i>	Secuencia de códigos de punto, rotulados como "PointCode" y el contenido indica el código de punto exacto.

2.1.1.1.5 routePriorityList (lista de prioridad de ruta)

Si el parámetro infoRequest está incluido y lo solicita, conforme se atraviesan los puntos de señalización (SP) cada uno de ellos añade la prioridad de la ruta al SP siguiente mediante routePriorityList.

<i>Parámetro</i>	<i>Código</i>
routePriorityList	10101100
<i>Contenido</i>	Secuencia de prioridad, rotulada como "Priority" con el contenido indica desconocido, primera elección, segunda elección, etc.

2.1.1.1.6 infoRequest (petición de información)

Este parámetro opcional, que sólo puede ser insertado por el SP que inicia la prueba, indica que el iniciador puede reconocer los mensajes MRVR ocasionados por un suceso del tipo de

routeTraceNew. El parámetro infoRequest indica cual es la información solicitada en caso de que deba enviarse el mensaje MRVR al iniciador. También puede indicar qué parámetros deben actualizarse conforme los mensajes MRVT atraviesan la red. Los valores vigentes pueden ser pointCode (bit 0 = 1) y/o pointCodeList (bit 1), y/o routePriorityList (bit 2).

<i>Parámetro</i>	<i>Código</i>
infoRequest	10001101
<i>Contenido</i>	
Cadena de bits que contiene uno o todos los valores indicados.	

2.1.1.1.7 returnUnknownParams (devolución de parámetros desconocidos)

Este parámetro opcional sólo puede ser insertado por el SP que inicia la prueba (sólo si el parámetro infoRequest también está incluido). Indica qué parámetros MRVT debe devolver el nodo siguiente si éste no reconoce dichos parámetros. Los parámetros MRVT no reconocidos deben ser copiados en el (nuevo) mensaje MRVR (routeTraceNew) si el nodo siguiente devuelve un MRVR (o en un mensaje MRVA del parámetro copyData si el iniciador le es desconocido). El bit 0 en returnUnknownParams indica un parámetro MRVT con valor de rótulo 15, bit 1 indica un parámetro MRVT con valor de rótulo 16, etc.

<i>Parámetro</i>	<i>Código</i>
returnUnknownParams	10001110
<i>Contenido</i>	
Cadena de bits que contiene cualquiera de los valores indicados.	

2.1.1.1.8 directRouteCheck (probar ruta directa)

Cuando este parámetro opcional es TRUE (VERDADERO) indica que los nodos siguientes deben verificar que disponen de una ruta con el iniciador de la prueba a través del SP del que han recibido el mensaje MRVT inicial.

<i>Parámetro</i>	<i>Código</i>
directRouteCheck	10001111
<i>Contenido</i>	
Booleano, TRUE indica que debe realizarse la prueba.	

2.1.1.2 Action Results (resultados de acciones)

No hay contenido en una indicación de devolución con éxito.

2.1.1.3 Action Errors (errores de acciones)

SpecificErrors son errores posibles que pueden producirse durante esta prueba, que son únicos de esta prueba. Estos errores específicos se añaden a los errores ya identificados en el servicio OM-CONFIRMED-ACTION y aparecen como parámetros del error de fallo de procesamiento.

2.1.1.3.1 failure (fallo)

failure indica una condición de fallo total, cuando ninguna ruta funciona correctamente. La mayoría de las veces se utilizará como una indicación de fallo a partir del punto que detecta el error y no invoca ningunas otras testRoute Actions. El SpecificError de fallo tiene un parámetro para indicar la

condición de error que produce el fallo. Este parámetro, `failureType`, se representa como una cadena de bits. El segundo parámetro ha de utilizarse cuando `failureType` indica el error `unknownInitiatingSP`. `traceSent` indica si se ha invocado o no `routeTrace` Event para informar sobre el rastreo. Es necesario indicar esto para este error ya que el nodo que detecta el error no puede enviar `routeTrace`, por lo que debe hacerlo el nodo anterior. `traceSent` es de tipo `BOOLEAN`. El tercer parámetro es opcional, estando presente si `failureType` es "unknownInitiating SP", `traceSent` es `FALSE` y el mensaje MRVT que inició la prueba contenía un parámetro `requestInfo` o un parámetro `returnUnknownParams` (o ambos).

<i>Error específico</i>	<i>Código</i>
<code>failure</code>	00000001

<i>Parámetro</i>	<i>Código</i>
<code>failureType</code>	10000000

<i>Parámetro</i>	<i>Código</i>
<code>traceSent</code>	10000001
<i>Contenido</i>	<i>Significado</i>
TRUE	Se envió la información del rastreo.
FALSE	No se envió la información del rastreo.

<i>Parámetro</i>	<i>Código</i>
<code>copyData</code>	10000100
<i>Contenido</i>	
Un OCTET STRING que contiene parámetros solicitados por el mensaje MRVT iniciador.	

2.1.1.3.2 `partialSuccess` (éxito parcial)

Esta indicación se da cuando por lo menos una invocación `testRoute Cnf Action` ha fallado y por lo menos una ha tenido éxito (por lo menos parcialmente). En este caso, cada tipo de error que se ha producido será anotado y enviado en la respuesta final. El formato y contenido de éxito parcial son iguales que para fracaso (`failure`).

<i>Error Específico</i>	<i>Código</i>
<code>partialSuccess</code>	00000010

2.1.2 `routeTrace` Event (suceso de rastreo de ruta)

`routeTrace` Event informa sobre el rastreo. La información de rastreo consiste en ninguno, uno o más códigos de punto, tales como el código de punto que detecta un error o la lista completa de códigos de puntos atravesados en una ruta. Este suceso se invoca a petición explícita del nodo originador (indicada por `traceRequested`, véase 2.1.1.1.2) o por fallo en cualquier punto de la ruta. Este suceso no es confirmado, por tanto no se esperan respuestas a esta invocación (no se esperan indicaciones de error ni de éxito).

<code>routeTrace</code> EVENT	<i>Temporizador</i> = 0	<i>Clase</i> = 4	<i>Código</i> = 00000010
-------------------------------	-------------------------	------------------	--------------------------

2.1.2.1 Event information (información de suceso)

2.1.2.1.1 success (éxito)

Cuando se termina con éxito, se incluye el rastreo de los códigos de punto (uno o más) de los puntos de señalización atravesados.

<i>Parámetro</i>	<i>Código</i>
success	10100000

2.1.2.1.2 detectedLoop (bucle detectado)

Cuando se detecta un bucle, se incluyen los códigos de punto (tres o más) contenidos en el bucle.

<i>Parámetro</i>	<i>Código</i>
detectedLoop	10100001

2.1.2.1.3 excessiveLengthRoute (ruta demasiado larga)

Cuando se encuentra una ruta demasiado larga (umbral rebasado), se incluye toda la ruta.

<i>Parámetro</i>	<i>Código</i>
excessiveLengthRoute	10100010

2.1.2.1.4 unknownDestination (destino desconocido)

Si el destino es desconocido, no se necesita información adicional, ya que el parámetro infoRequest no fue incluido en la petición testRoute CNF-ACTION.

2.1.2.1.5 routeInaccessible (ruta inaccesible)

Se incluye el código de punto del nodo en el que la ruta ha resultado inaccesible.

<i>Parámetro</i>	<i>Código</i>
routeInaccessible	10000100

2.1.2.1.6 processingFailure (fallo de procesamiento)

Si hay un fallo de procesamiento, no se necesita información adicional.

<i>Parámetro</i>	<i>Código</i>
processingFailure	10000101

2.1.2.1.7 unknownInitiatingSP (SP iniciador desconocido)

Se incluye el código de punto del nodo que detecta el punto de señalización iniciador desconocido.

<i>Parámetro</i>	<i>Código</i>
unknownInitiatingSP	10000110

2.1.2.1.8 timerExpired (temporizador expirado)

Se incluye el o los códigos de punto de los que no se ha recibido resultado para testRoute Action.

Parámetro	Código
timerExpired	10100111

2.1.2.1.9 sPNotAnSTP (SP no es un STP)

Si el punto de señalización intermedio que recibe un mensaje MRVT no tiene la función de transferencia MTP, se incluye la lista de puntos de señalización atravesados para alcanzar este punto de señalización.

El valor "sPNotAnSTP" de failureType puede también significar que el punto de señalización intermedio que recibe el mensaje MRVT no está autorizado a transferir mensajes recibidos de quien envía la MRVT, cuya etiqueta OPC de la MTP es la del iniciador de la prueba y la DPC la del destinatario de la misma.

Parámetro	Código
sPNotAnSTP	10101000

2.1.2.1.10 maxNrMRVTestsAlready (alcanzado número máximo de pruebas MRV)

Este informe es utilizado por el punto de señalización que recibe el mensaje MRVT si en el SP se alcanza el número máximo, n_r , de pruebas MRV. Si el mensaje MRVT iniciador (testRoute) no contiene al parámetro infoRequest, se informa "processingFailure" (fallo de procesamiento), véase 2.1.2.1.6.

2.1.3 routeTraceNew (nuevo rastreo de ruta)

Este informe se utiliza si la acción testRoute iniciador contiene el parámetro infoRequest.

SUCESO routeTraceNew	Temporizador = 0	Clase = 4	Código = 00000100
----------------------	------------------	-----------	-------------------

2.1.3.1 Información sobre el suceso

2.1.3.1.1 success (éxito)

Cuando se termina con éxito, los códigos de punto de los puntos de señalización atravesados se incluyen en pointCodeList (que se ha copiado del parámetro pointCodesTraversed de la acción testRoute).

routePriorityList se copia de la acción testRoute si está presente y así lo solicita el parámetro infoRequest.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

2.1.3.1.2 detectedLoop (bucle detectado)

Cuando se detecta un bucle, se incluyen en pointCodeList los códigos de punto (tres o más) de los puntos de señalización contenidos en el bucle.

routePriorityList se copia de la acción testRoute si está presente y así lo solicita el parámetro infoRequest.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

2.1.3.1.3 excessiveLengthRoute (ruta demasiado larga)

Si este error tiene lugar, se copia toda la ruta del parámetro pointCodesTraversed de la acción testRoute en el parámetro pointCodeList.

routePriorityList se copia de la acción testRoute si está presente y así lo solicita el parámetro infoRequest.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

2.1.3.1.4 unknownDestination (destino desconocido)

Este caso es equivalente al unknownDestination descrito en 2.1.2.1.4. Si el parámetro infoRequest de la acción iniciadora testRoute lo solicita, se copia el parámetro pointCodesTraversed de la acción testRoute en pointCodeList.

routePriorityList se copia de la acción testRoute si está presente y así lo solicita el parámetro infoRequest.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

2.1.3.1.5 routeInaccessible (ruta inaccesible)

Si este suceso informa de un solo SP inaccesible, su código de punto se incluye en pointCode.

Si se informa de más de un SP inaccesible (y por tanto, la acción iniciadora testRoute indica que el originador puede aceptarlo), la lista de todos los SP inaccesibles se incluye en pointCodeList.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

2.1.3.1.6 processingFailure (fallo de procesamiento)

Si la prueba no puede realizarse debido a condiciones locales, el suceso informa de un fallo de procesamiento. Ello incluye el rechazo de la acción testRoute por parte de la SCCP o la TC en un SP distante.

Si la acción testRoute del parámetro infoRequest estaba presente y tenía el bit 0 puesto a 1, el código de punto del SP en el que fracasó la prueba se incluye en el parámetro pointCode.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

2.1.3.1.7 unknownInitiatingSP (SP iniciador desconocido)

En el parámetro pointCode se incluye el código de punto del SP que detecta al iniciador desconocido.

Si el resultado iniciador testRoute contiene un parámetro copyData, éste se copia en el parámetro copyData de routeTraceNew.

2.1.3.1.8 timerExpired (temporizador expirado)

En pointCodeList se incluyen los códigos de punto de los SP de los que no se ha recibido ningún resultado de las acciones testRoute.

2.1.3.1.9 sPNotAnSTP (SP no es un STP)

Este error tiene lugar si el SP intermedio no tiene la función STP o se sabe que no está autorizado para transferir mensajes desde el iniciador de la prueba al destinatario de la misma.

El parámetro pointCodesTraversed de la acción iniciadora testRoute se copia en pointCodeList.

routePriorityList se copia de la acción testRoute si está presente y así lo solicita el parámetro infoRequest.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

2.1.3.1.10 indirectRoute (ruta indirecta)

Este informe se utiliza si se ha solicitado la prueba de ruta directa y el SP que recibe el mensaje MRVT no dispone de ruta con el iniciador de la prueba directamente a través del emisor del mensaje MRVT. La identidad del emisor del mensaje MRVT iniciador se incluye en el parámetro pointCode y la identidad del SP que detecta la falta de ruta directa figura en el OPC de la etiqueta MTP del mensaje MRVR.

2.1.3.1.11 maxNrMRVTestsAlready (número máximo de pruebas MRVT alcanzado)

Este informe es utilizado por el punto de señalización que recibe el mensaje MRVT si en el SP se ha alcanzado el número máximo (n_r) de pruebas MRV.

Si el parámetro infoRequest de la acción testRoute estuviera presente y tuviera el bit 0 puesto a 1, el código de punto del SP en el que fracasó la prueba se incluye en el parámetro pointCode.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3 Parte control de la conexión de señalización (SCCP)

3.1 ASE de la prueba de verificación del encaminamiento SCCP (SRVT)

Estas funciones específicas de SRVT se definen en 3.2.2/Q.753. La prueba SRV iniciada en el origen de prueba hace que se utilice una primitiva OM-CONFIRMED-ACTION del usuario OMASE a la OMASE, que incluye como parámetro la instrucción testRoute. Si se solicita el rastreo de las rutas o hay una avería, se invoca la primitiva OM-EVENT-REPORT en el originador de la prueba desde OMASE, que incluye como parámetro el suceso routeTrace.

La instrucción testRoute se especifica utilizando CNF-ACTION definido en la figura 3, y el suceso routeTrace se especifica mediante EVENT definido en la figura 3.

ObjectClass indica tablas de traducción de título global SCCP y ObjectInstance contiene el indicador de título global y el título global probado. El indicador de título global se codifica como se define en el indicador de dirección SCCP. testRoute Action (SRVT) utiliza el mensaje BEGIN y el resultado (SRVA) se devuelve en END. routeTrace Event (SRVR) utiliza un mensaje BEGIN con fin preacordado.

3.1.1 testRouteAction (acción probar ruta)

testRoute Action se invoca para iniciar una prueba de verificación de encaminamiento SCCP. En el nodo iniciador, esta invocación es solicitada por la Administración a través del usuario MIS o una interfaz local, mediante el proceso de gestión OMAP y el usuario OMASE. En los nodos siguientes, la acción es solicitada implícitamente al recibirse una invocación testRoute Action. Una respuesta positiva indica la terminación exitosa de la prueba en el punto en que se invocó e, implícitamente, en todos los puntos siguientes donde se invocó. Se devuelve una indicación de fracaso para informar que la prueba falló en este nodo o en otro siguiente.

testRoute CNF-ACTION	Temporizador = T2	Clase = 1	Código = 00000001
----------------------	-------------------	-----------	-------------------

3.1.1.1 testRoute action arguments (argumentos de la acción probar ruta)

3.1.1.1.1 initiatingSP (punto de señalización iniciador)

initiatingSP identifica al iniciador de la prueba. Es del tipo PointCode.

<i>Parámetro</i>	<i>Código</i>
initiatingSP	10000000
<i>Contenido</i>	
Bit 0 contiene el primer bit del código de punto.	
Bit 1 contiene el segundo bit del código de punto, etc.	

3.1.1.1.2 traceRequested (rastreo solicitado)

traceRequested indica que se debe informar sobre el rastreo de todas las rutas utilizadas para llegar al destino. Es de tipo BOOLEAN.

<i>Parámetro</i>	<i>Código</i>
traceRequested	10000001
<i>Contenido</i>	<i>Significado</i>
TRUE (= 1)	Se solicitó el rastreo, devolución de información de éxito/fracaso del rastreo.
FALSE (= 0)	No se solicitó rastreo, devolución de información sólo si fracasa el rastreo.

3.1.1.1.3 threshold (umbral)

El originador fija un nivel de umbral máximo para los puntos de señalización de traducción (TSP, *translation signalling points*) que pueden cruzarse en el transcurso de la prueba (incluido el iniciador, si es un nodo relevador SCCP). Esto ayuda a detectar rutas demasiado largas. Este umbral es un número entero de puntos de señalización, de modo que es del tipo INTEGER.

<i>Parámetro</i>	<i>Código</i>
threshold	10000010

3.1.1.1.4 pointCodesTraversed (códigos de puntos atravesados)

Cuando cada punto de señalización de traducción es atravesado, añade su propio código de punto a la lista de códigos de puntos atravesados. Esto ayuda a detectar bucles y es también una información útil en caso de fallo o si se solicita el rastreo de una ruta. Es una lista de códigos de puntos del tipo PointCodeList. Esta PointCodeList podrá estar vacía.

<i>Parámetro</i>	<i>Código</i>
pointCodesTraversed	10100011
<i>Contenido</i>	
Secuencia de códigos de puntos rotulados como "PointCode" y el contenido indica el código de punto exacto.	

3.1.1.1.5 formIndicator (indicador de forma)

formIndicator identifica la forma del mensaje SRVT, es decir, pedir, verificar o comparar. Es del tipo INTEGER, con los valores definidos a continuación.

<i>Parámetro</i>	<i>Código</i>
formIndicator	10000100
<i>Contenido</i>	
Valor 0 = Comparar. Valor 1 = No comparar.	

3.1.1.1.6 mtpBackwardRoutingRequested (solicitado encaminamiento hacia atrás por la MTP)

mtpBackwardRoutingRequested identifica si se requiere encaminamiento hacia atrás por la MTP hacia OPC para el éxito de la prueba. Es del tipo BOOLEAN.

<i>Parámetro</i>	<i>Código</i>
mtpBackwardRoutingRequested	10000101
<i>Contenido</i>	
TRUE (= 1) Encaminamiento solicitado. FALSE (= 0) Encaminamiento no solicitado.	

3.1.1.1.7 testInitiatorGT (título global de iniciador de la prueba)

testInitiatorGT identifica el indicador de título global y el título global del iniciador. Es del tipo OCTET STRING.

<i>Parámetro</i>	<i>Código</i>
testInitiatorGT	10000110
<i>Contenido</i>	
Octeto 1 bits 3 a 6 = Indicador de título global. Octetos 2, 3, ... = Título global del iniciador.	

3.1.1.1.8 destinationPC (código de punto de destino)

destinationPC identifica el código de punto de destino (PPC o TPC). Es del tipo PointCode.

<i>Parámetro</i>	<i>Código</i>
destinationPC	10000111
<i>Contenido</i>	
Bit 0 contiene el primer bit del código de punto.	
Bit 1 contiene el segundo bit del código de punto, etc.	

3.1.1.1.9 destinationSSN (número de subsistema de destino)

destinationSSN identifica al número del subsistema de destino. Es del tipo OCTET STRING.

<i>Parámetro</i>	<i>Código</i>
destinationSSN	10001000
<i>Contenido</i>	
Bit 0 contiene el primer bit del número de subsistema.	
Bit 1 contiene el segundo bit del número de subsistema, etc.	

3.1.1.1.10 backupDPC (código de punto de destino de respaldo)

backupDPC identifica el código de punto de destino de respaldo (SPC). Es del tipo PointCode.

<i>Parámetro</i>	<i>Código</i>
backupDPC	10001001
<i>Contenido</i>	
Bit 0 contiene el primer bit del código de puntos.	
Bit 1 contiene el segundo bit del código de punto, etc.	

3.1.1.1.11 backupSSN (número de subsistema de respaldo)

backupSSN identifica el número de subsistema de destino de respaldo. Es del tipo OCTET STRING.

<i>Parámetro</i>	<i>Código</i>
backupSSN	10001010
<i>Contenido</i>	
Bit 0 contiene el primer bit del número de subsistema.	
Bit 1 contiene el segundo bit del número de subsistema, etc.	

3.1.1.1.12 originalGT (título global original)

El campo "originalGT" está presente únicamente en un mensaje SRVT si la traducción de un GT en la dirección de la parte llamada produce o ya ha producido un GT de sustitución.

En este caso, el campo que debe enviarse en un mensaje SRVT es de la forma siguiente:

- i) si el mensaje SRVT que va a enviarse no es el formulario de comparación y la prueba se inicia al recibirse un mensaje SRVT que contiene un campo originalGT, se copia completamente el campo;

- ii) en los demás casos, el campo originalGT enviado es el GT de la dirección de la parte llamada del mensaje SRVT antes de la traducción.

El campo se utiliza como GT de la dirección de la parte que llama en cualquier mensaje SRVR enviado, y, para el formulario de comparación del mensaje SRVT, es utilizado por el punto de señalización de traducción (TSP) asociado que lo recibe para verificar que su traducción da el GT en el campo de dirección de la parte llamada del mensaje de comparación SRVT recibido.

El tipo de GT original es un título global.

<i>Parámetro</i>	<i>Código</i>
originalGT	10001011
<i>Contenido</i>	
Octeto 1 bits 3 a 6 = Indicador de título global.	
Octetos 2, 3,... = Título global original.	

3.1.1.1.13 inputGT (título global de entrada)

inputGT, que sólo se utiliza en la forma comparar del mensaje SRVT, identifica el GT y GTI de prueba previamente a la traducción en un TSP. Es del tipo OCTET STRING.

<i>Parámetro</i>	<i>Código</i>
inputGT	10010000
<i>Contenido</i>	
Bits 3 a 6 del octeto 1 = Indicador de título global (GTI).	
Octetos 2, 3, ... = Título global de entrada al TSP.	

3.1.1.1.14 infoRequest (petición de información)

Este parámetro opcional, que sólo puede ser insertado por el SP que inicia la prueba, indica que el iniciador puede reconocer los mensajes SRVR ocasionados por un suceso del tipo de routeTraceNew. El parámetro infoRequest indica cual es la información solicitada en caso de que deba enviarse el mensaje SRVR al iniciador. También puede indicar qué parámetros deben actualizarse conforme los mensajes SRVT atraviesan la red. Los valores vigentes pueden ser pointCode (bit 0 = 1) y/o pointCodeList (bit 1).

<i>Parámetro</i>	<i>Código</i>
infoRequest	10001101
<i>Contenido</i>	
Cadena de bits que contiene uno o todos los valores indicados.	

3.1.1.1.15 returnUnknownParams (devolución de parámetros desconocidos)

Este parámetro opcional sólo puede ser insertado por el SP que inicia la prueba (sólo si el parámetro infoRequest también está incluido). Indica qué parámetros SRVT debe devolver el nodo siguiente si éste no reconoce dichos parámetros. Los parámetros SRVT no reconocidos deben ser copiados en el (nuevo) mensaje SRVR (routeTraceNew) si el nodo siguiente devuelve un SRVR (o en un mensaje SRVA del parámetro copyData si el iniciador le es desconocido). El bit 0 en returnUnknownParams indica un parámetro SRVT con valor de rótulo 15, bit 1 indica un parámetro SRVT con valor de rótulo 16, etc.

<i>Parámetro</i>	<i>Código</i>
returnUnknownParams	10001110
<i>Contenido</i>	
Cadena de bits que contiene cualquiera de los valores indicados.	

3.1.1.2 Action Results (resultados de acciones)

No hay contenido en una indicación de devolución con éxito.

3.1.1.3 Action Errors (errores de acciones)

SpecificErrors son posibles errores que pueden producirse durante esta prueba que son únicos a esta prueba. Estos errores específicos se añaden a los errores ya identificados en el servicio OM-CONFIRMED-ACTION y aparecen como parámetros para el error de fallo de procesamiento.

3.1.1.3.1 failure (fallo)

failure indica una condición de fallo, cuando una traducción no pudo efectuarse con éxito, o fue incorrecta. La mayoría de las veces se utilizará como indicación de fallo a partir del punto que detecta el error y no invoca ninguna otra testRoute Actions. SpecificError de fallo tiene un parámetro para indicar la condición de error que produce el fallo. Este parámetro, failureType, se representa como una cadena de bits. Además, debe utilizarse el segundo parámetro cuando failureType indica el error Unknown Initiating SP. traceSent indica si se ha invocado o no routeTrace Event para informar sobre el rastreo. Es necesario indicarlo para este error, pues el nodo que detecta el error no puede enviar routeTrace, por lo que debe hacerlo el nodo anterior. traceSent es un tipo de BOOLEAN y es facultativo.

<i>Error específico</i>	<i>Código</i>
failure	00000001

<i>Parámetro</i>	<i>Código</i>
failureType	10000000

<i>Parámetro</i>	<i>Código</i>
traceSent	10000001
<i>Contenido</i>	<i>Significado</i>
TRUE	Se envió información de rastreo.
FALSE	No se envió información de rastreo.

3.1.1.3.2 partialSuccess (éxito parcial)

Esta indicación se presenta cuando por lo menos una invocación testRoute Cnf Action ha fracasado y por lo menos una ha tenido éxito (por lo menos parcialmente). En este caso, cada tipo de error que se ha producido se anotará y enviará en la respuesta final. El formato y el contenido de éxito parcial son iguales al de fallo (failure).

<i>Error específico</i>	<i>Código</i>
partialSuccess	00000010

3.1.2 routeTrace Event (suceso de rastreo de ruta)

routeTrace Event informa sobre el rastreo. La información sobre rastreo consiste en uno o más códigos de punto, tal como la lista completa de códigos de punto de traducción atravesados a lo largo de una ruta. Este suceso se invoca a petición explícita del nodo originador (indicada por traceRequested, véase 3.1.1.1.2) o por fallo en cualquier punto de la ruta. Este suceso es no confirmado, por lo que no se esperan respuestas a esta invocación (no se esperan indicaciones de error ni de éxito).

routeTrace EVENT	Temporizador = 0	Clase = 4	Código = 00000010
------------------	------------------	-----------	-------------------

3.1.2.1 Event information (información de suceso)

3.1.2.1.1 success (éxito)

Cuando se termina satisfactoriamente, se incluye el rastreo del código o códigos de puntos de los nodos relevadores SCCP atravesados.

Parámetro	Código
success	10100000

3.1.2.1.2 detectedLoop (bucle detectado)

Cuando se detecta un bucle, se incluyen los tres o más códigos de punto contenidos en el bucle.

Parámetro	Código
detectedLoop	10100001

3.1.2.1.3 excessiveLengthRoute (ruta demasiado larga)

Cuando se encuentra una ruta demasiado larga (umbral excedido), se incluye toda la ruta.

Parámetro	Código
excessiveLengthRoute	10100010

3.1.2.1.4 unknownDestination (destino desconocido)

Si el destino es desconocido, no se necesita información adicional. Para la SRVT, esto se relaciona con el caso cuando no existen datos de traducción para el GTI + GT.

Parámetro	Código
unknownDestination	10000011

3.1.2.1.5 routeInaccessible (ruta inaccesible)

Se incluye el código de punto del nodo en el que la ruta ha resultado inaccesible.

Parámetro	Código
routeInaccessible	10000100

3.1.2.1.6 processingFailure (fallo de procesamiento)

Si hay un fallo de procesamiento, no se necesita información adicional.

<i>Parámetro</i>	<i>Código</i>
processingFailure	10000101

3.1.2.1.7 unknownInitiatingSP (SP iniciador desconocido)

Se incluye el código de punto del nodo que detecta el SP iniciador desconocido.

<i>Parámetro</i>	<i>Código</i>
unknownInitiatingSP	10000110

3.1.2.1.8 timerExpired (temporizador expirado)

Se incluye el o los códigos de punto de los que no se ha recibido resultado para testRoute Action.

<i>Parámetro</i>	<i>Código</i>
timerExpired	10100111

3.1.2.1.9 wrongSP (punto de señalización erróneo)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al punto de señalización no válido.

<i>Parámetro</i>	<i>Código</i>
wrongSP	10101000

3.1.2.1.10 incorrectTranslation-Primary (traducción-destino primario incorrecto)

Se incluye la lista completa de los puntos de señalización de traducción atravesados en la ruta al destino primario incorrecto.

<i>Parámetro</i>	<i>Código</i>
incorrectTranslation-Primary	10101001

3.1.2.1.11 incorrectTranslation-Secondary (traducción-destino secundario incorrecto)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al destino secundario incorrecto.

<i>Parámetro</i>	<i>Código</i>
incorrectTranslation-Secondary	10101010

3.1.2.1.12 incorrectTranslation-Intermediate (traducción-punto intermedio incorrecto)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al punto intermedio incorrecto.

<i>Parámetro</i>	<i>Código</i>
incorrectTranslation-Intermediate	10101011

3.1.2.1.13 notPrimaryDestination (destino primario no válido)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al destino primario no válido.

<i>Parámetro</i>	<i>Código</i>
notPrimaryDestination	10101100

3.1.2.1.14 notSecondaryDestination (destino secundario no válido)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al destino secundario no válido.

<i>Parámetro</i>	<i>Código</i>
notSecondaryDestination	10101101

3.1.2.1.15 notRecognizedPrimary (destino primario no reconocido)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al destino secundario.

<i>Parámetro</i>	<i>Código</i>
notSecondaryDestination	10101110

3.1.2.1.16 notRecognizedSecondary (destino secundario no reconocido)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al destino primario.

<i>Parámetro</i>	<i>Código</i>
notRecognizedSecondary	10101111

3.1.2.1.17 routingProblem (problema de encaminamiento)

Se incluye la lista completa de puntos de señalización de traducción atravesados en la ruta al posible problema de encaminamiento. Esto se produce cuando no se ha reconocido el código de punto resultante de la traducción.

<i>Parámetro</i>	<i>Código</i>
routingProblem	10110000

3.1.3 Suceso routeTraceNew (nuevo rastreo de ruta)

Este informe se utiliza si la acción de testRoute iniciadora contiene un parámetro infoRequest.

SUCESO routeTraceNew	Temporizador = 0	Clase = 4	Código = 00000100
----------------------	------------------	-----------	-------------------

3.1.3.1 Información del suceso

La información incluida es como la que se incluye para routeTrace, pero con el parámetro "resultado" ("result") indicando el resultado, estando los datos asociados al resultado incluidos en los parámetros opcionales. El parámetro copyData contiene, si está presente, aquellos parámetros de la acción testRoute iniciadora que no eran comprensibles y que el returnUnknownParams de la acción

testRoute solicitó que fueran devueltos. Se define un resultado adicional de routeTrace para routeTraceNew, se trata del maxNrSRVRTestsAlready.

3.1.3.1.1 success (éxito)

Cuando se termina con éxito, se incluye en pointCodeList el rastreo de los códigos de punto (uno o más) de los SP atravesados (se copian del parámetro pointCodesTraversed de la acción testRoute).

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.2 detectedLoop (bucle detectado)

Cuando se detecta un bucle, se incluyen en pointCodeList los códigos de punto de los SP (tres o más) contenidos en el bucle.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.3 excessiveLengthRoute (ruta demasiado larga)

Si este error tiene lugar, se copia toda la ruta del parámetro pointCodesTraversed de la acción testRoute en el parámetro pointCodeList.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.4 unknownDestination (destino desconocido)

Si el parámetro infoRequest de la acción testRoute iniciadora los solicita, se copia el parámetro pointCodesTraversed de la acción testRoute en pointCodeList.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.5 routeInaccessible (ruta inaccesible)

Si este suceso informa de un único SP inaccesible, su código de punto se incluye en pointCode.

En caso de que se informe de más de un SP inaccesible (y por tanto la acción testRoute iniciadora informa de que el originador puede aceptarlo), la lista de todos los SP inaccesibles se incluye en pointCodeList.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.6 processingFailure (fallo de procesamiento)

Si la prueba no puede realizarse debido a condiciones locales, el suceso informa de un fallo de procesamiento. Ello incluye el rechazo de la acción testRoute por parte de la SCCP o la TC en un SP distante.

Si la acción testRoute del parámetro infoRequest estaba presente y tenía el bit 0 puesto a 1, el código de punto del SP en el que fracasó la prueba se incluye en el parámetro pointCode.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.7 unknownInitiatingSP (SP iniciador desconocido)

En el parámetro pointCode se incluye el código de punto del SP que detecta al iniciador desconocido.

Si el resultado iniciador testRoute contiene un parámetro copyData, éste se copia en el parámetro copyData del routeTraceNew.

3.1.3.1.8 timerExpired (temporizador expirado)

En pointCodeList se incluyen los códigos de punto de los SP de los que no se ha recibido ningún resultado de las acciones testRoute.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.9 wrongSP (SP equivocado)

El parámetro pointCodesTraversed de la acción testRoute iniciadora se copia en el pointCodeList.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.10 incorrectTranslation-Primary (traducción-destino primario incorrecto)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al destino primario incorrecto.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.11 incorrectTranslation-Secondary (traducción-destino secundario incorrecto)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al destino secundario incorrecto.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.12 incorrectTranslation-Intermediate (traducción-destino intermedio incorrecto)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al punto intermedio incorrecto.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.13 notPrimaryDestination (destino primario no válido)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al destino primario no válido.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.14 notSecondaryDestination (destino secundario no válido)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al destino secundario no válido.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.15 notRecognizedPrimary (destino primario no reconocido)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al destino secundario.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.16 notRecognizedSecondary (destino secundario no reconocido)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al destino primario.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.17 routingProblem (problema de encaminamiento)

En el pointCodeList se incluye la lista completa de los SP de traducción atravesados en la ruta al posible problema de encaminamiento. Esto tiene lugar cuando no se ha reconocido el código de punto resultante de la traducción.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

3.1.3.1.18 maxNrSRVTestsAlready (número máximo de pruebas SRVT alcanzado)

Este informe es utilizado por el punto de señalización que recibe el mensaje SRVT si en el SP se alcanza el número máximo (n_{SRVT}) de pruebas SRV.

Si el parámetro infoRequest de la acción testRoute estuviera presente y tuviera el bit 0 puesto a 1, el código de punto del SP en el que fracasó la prueba se incluye en el parámetro pointCode.

Si en la acción iniciadora testRoute existen parámetros que no son comprensibles y ésta contiene un parámetro returnUnknownParams que los solicita, dichos parámetros se copian en el parámetro copyData.

4 Gestión de circuitos

4.1 Elemento de servicio de aplicación de la prueba de validación de circuitos (CVT)

El elemento de servicio de aplicación (ASE) de la prueba de validación de circuitos (CVT) proporciona los servicios a los que se tiene acceso por la OM-CONFIRMED-ACTION descrita en la figura 3. Utiliza una instancia basada en una clase de objeto de gestión de circuitos definida en la Recomendación Q.751. BaseManagedObjectClass indica cvt-Cic-Tables-1992 y BaseManagedObjectInstance identifica CktGrpInfo (información de grupo de circuitos que señala al identificador predefinido para el circuito y su grupo acordado entre centrales en los extremos del grupo de circuitos) y su código de identificación de circuito (CIC) en el punto de señalización de emisión.

4.1.1 cktValidTest CnfAction (acción confirmada prueba de validación de circuito)

La petición de prueba de validación de circuito y la devolución consiguiente de la respuesta de validación de circuito se corresponden con una acción confirmada. La acción es la petición de la prueba en el extremo lejano.

cktValidTest CNF-ACTION	Temporizador = T_c	Clase = 1	Código = 00000001
-------------------------	----------------------	-----------	-------------------

En 4.2/Q.753 pueden verse posibles valores de T_c y del temporizador.

```
cktValidTest CNF-ACTION ::=
{
  ACTIONARG SEQUENCE {
    requestingSP          RequestingSP,
    timer                 Timer OPTIONAL,
    ...}

  ACTIONRESULT           Success
  SPECIFICERRORS        { failure }
  CODE                   3
}
-- Timer =  $T_c$ , class = 1
```

4.1.2 Action Arguments (argumentos de acción)

El punto de señalización solicitante es el código de punto del punto de señalización que inicia el procedimiento de prueba. Es del tipo Octet String como se indica a continuación.

```
RequestingSP ::= OCTET STRING
```

4.1.3 Action Results (resultados de acción)

Los resultados de acciones se devuelven en un componente de devolución de resultado cuando tienen éxito. El contenido de los dos parámetros debe definirse basado en el procedimiento CVT.

```
Success ::= SEQUENCE
{
  cktGrpInfo [0] IMPLICIT CktGrpInfo,
  cICName    [1] IMPLICIT OCTET STRING OPTIONAL,
  ...}

```

Obsérvese que CktGrpInfo se define como OCTET STRING.

4.1.4 Specific Error (error específico)

El error específico indica el fracaso y el motivo del mismo. El contenido de los dos parámetros ha de definirse basado en el procedimiento CVT.

Failure ::= SPECIFIC-ERROR

```
{
  PARAMETER SEQUENCE   {cktGrpInfo  [0] IMPLICIT CktGrpInfo,
                        cICName      [1] IMPLICIT OCTET STRING OPTIONAL,
                        ...}
  CODE                 3
}
```

Obsérvese que CktGrpInfo se define como OCTET STRING.

Los motivos del fallo de la CVT son:

- a) no asignación de CIC en el extremo distante;
- b) datos erróneos para el circuito en el extremo cercano;
- c) no recepción de tono válido en el extremo cercano;
- d) expiración del temporizador de prueba global T_c antes de la recepción CVR;
- e) recepción de mensaje CVR antes de alcanzar la sincronización en la prueba de la configuración de bits;
- f) expiración de T_c antes de alcanzar la sincronización en la prueba de la configuración de bits;
- g) la recepción de la configuración de bits está todavía en curso cuando expira T_c ;
- h) la recepción de la configuración de bits está todavía en curso cuando se recibe el mensaje CVR;
- i) la recepción del tono está en curso cuando expira T_c ;
- j) la recepción del tono está en curso cuando se recibe el mensaje CVR;
- k) no concordancia entre el CIC del extremo cercano y el CIC del extremo distante (comprobación en el extremo cercano al recibirse el mensaje CVR);
- l) recepción de mensaje CVR indicando fallo:
 - no asignación de CIC en el extremo distante;
 - datos erróneos para el circuito en el extremo cercano;
 - indisponibilidad de las características de grupo en el extremo distante;
- m) fallo, no especificado.

5 Capacidades de transacción (TC)

En estudio.

La respuesta que se incluye en la OMAP a la difusión de N-STATE y N-PCSTATE iniciadas por la SCCP también queda en estudio.

6 Definiciones generales

6.1 Objetos y operaciones

La OMAP realiza pruebas en objetos tales como las tablas de encaminamiento MTP y SCCP. Estos objetos se describen en este documento como "clase de objeto" y son identificados por un identificador de objeto que especifica esta Recomendación y el tipo de objeto. Esta estructura se muestra a continuación para los identificadores de objeto OMAP mtp-Routing-Tables, sccp-Routing-Tables, y cvt-Cic-Tables.

oMAP	OBJECT IDENTIFIER ::= { itu-t recommendation q 754 }
mtp-Routing-Tables-1992	OBJECT IDENTIFIER ::= { oMAP 0 }
sccp-Routing-Tables-1992	OBJECT IDENTIFIER ::= { oMAP 1 }
cvt-Cic-Tables-1992	OBJECT IDENTIFIER ::= { oMAP 5 }

La clase de objeto de tablas de encaminamiento MTP es 0011857200 (hexadecimal), para las tablas de encaminamiento SCCP es 0011857201 (hexadecimal) y para las tablas CVT CIC es 0011857205 (hexadecimal). Véanse las Recomendaciones X.680 y X.690.

Los cuadros 1 y 2 muestran las primitivas OM; en la figura 1 aparecen las operaciones OMAP obtenidas de CMIP (ISO/CEI 9596) y la figura 3 ilustra una sintaxis formal de OMASE.

Operaciones definidas actualmente	
0	eventReport
7	confirmedAction

**Figura 1/Q.754 – Operaciones OMAP
obtenidas de CMIP**

6.2 Primitivas y procedimientos del protocolo OMASE

6.2.1 Consideraciones generales

El protocolo OMASE utiliza el servicio TC definido en la Recomendación Q.771. Invoke ID y Dialogue ID corresponden a los definidos para el servicio TC.

OMASE se modela mediante una máquina de protocolo (denominada OMPM). En lo que sigue, la unidad de datos de protocolo de aplicación (APDU, *application protocol data unit*) hace referencia al contenido de la primitiva o primitivas cursadas entre OMASE y TC.

En la figura A.1 aparece el modelo, que incluye TC y SCCP. El OMPM reside en la OMASE. La figura A.2 muestra un ejemplo de instancias de primitivas particulares en una prueba MRV (pero sin OM-EVENT-REPORT).

6.2.2 OM-EVENT-REPORT (OM-información de suceso)

6.2.2.1 Primitiva de servicio

En el cuadro 1 se define la primitiva OM-EVENT-REPORT utilizada entre el usuario OMASE y OMASE.

El suceso específico que se ha producido se interpreta en el contexto de la clase objeto especificada.

Cuadro 1/Q.754 – Parámetros OM-EVENT-REPORT

Nombre de parámetro	pet./ind.
CallingPartyAddress	M
CalledPartyAddress	M
DialogueID	M
InvokeID	M
ManagedObjectClass	M
ManagedObjectInstance	M
EventType	M
EventTime	O
EventInfo	O

Definiciones de parámetros

CallingPartyAddress: Definida en la dirección llamante de 2.2/Q.711.

CalledPartyAddress: Definida en la dirección llamada de 2.2/Q.711. Estas direcciones sirven para identificar la OMAP en los puntos de señalización llamante y llamado respectivamente. Para MRVT, pueden tener la forma de un código de punto más un número de subsistema (OMAP), y para SRVT tienen una forma adecuada para el tipo de encaminamiento de SCCP utilizado en la prueba.

DialogueID: Definido en las Recomendaciones Q.771-Q.775. Corresponde a la Transaction ID, definida en la Recomendación Q.772.

InvokeID: Definido en la Recomendación Q.772.

ManagedObjectClass: Identifica la clase de objetos para la cual se define este suceso.

ManagedObjectInstance: Identifica la instancia de objeto sobre la que informa el suceso.

EventType: Especifica el suceso particular informado por la instancia de objeto.

EventTime: Especifica la hora en la que se generó el suceso.

EventInfo: Proporciona información adicional específica del suceso.

6.2.2.2 Procedimiento de informe de suceso

6.2.2.2.1 Recepción de la petición OM-EVENT-REPORT

El procedimiento de informe de suceso se inicia al recibirse la primitiva de petición OM-EVENT-REPORT. Cuando sucede esto, la OMPM construye la APDU que solicita la operación eventReport y transmite la APDU mediante los servicios TC-INVOKE y TC-BEGIN.

La primitiva de petición TC-INVOKE contiene los siguientes parámetros y valores:

- Dialogue ID – Definido por el usuario OMASE.
- Invoke ID – Definido por el usuario OMASE.
- Operación – Fijado a eventReport.
- Clase – Fijado a 4.

- Parámetros – Los que siguen a la palabra "PARAMETER" en la definición de eventReport. El valor del parámetro eventType especifica la acción que va a llevarse a cabo; debe indicar routeTrace para los procedimientos definidos actualmente.
- Tiempo de inactividad – Fijado a 0 para MRVT y SRVT.

La primitiva de petición TC-BEGIN utiliza los siguientes parámetros y valores:

- Dirección de destino – Como la recibida en la primitiva de petición OM-EVENT-REPORT CalledPartyAddress.
- Dirección de origen – Como la recibida en la primitiva de petición OM-EVENT-REPORT CallingPartyAddress.
- Dialogue ID – Como en TC-INVOKE.

La primitiva de petición N-UNITDATA emitida al SCCP al recibirse estas primitivas de petición TC debe contener el parámetro de control de secuencia puesto a "secuencia garantizada", y el parámetro de opción de devolución debe ponerse a "en caso de error descartar mensaje". Véase 2.2.2/Q.711.

Una vez transmitida la APDU, el OMPM finaliza el diálogo mediante la primitiva de petición TC-END con los parámetros Dialogue ID y Termination, indicando este último "fin previamente acordado".

6.2.2.2 Recepción de TC-BEGIN con indicación TC-INVOKE

Al recibir una APDU bien constituida y que solicita la operación eventReport de las primitivas indicación TC-BEGIN y TC-INVOKE, el OMPM emite una primitiva indicación OM-EVENT-REPORT. Si la APDU no está bien constituida, OMPM la descarta.

OMPM finaliza el diálogo con una primitiva petición TC-END con los parámetros Dialogue ID y Termination, indicando este último "fin previamente acordado".

6.2.2.3 Recepción de TC-BEGIN con indicación TC-L-REJECT

En este caso, el OMPM emite una primitiva petición TC-END con los parámetros Dialogue ID y Termination, indicando este último "fin previamente acordado".

6.2.2.4 Recepción de la indicación TC-P-ABORT

En este caso, OMPM ignora TC-P-ABORT.

6.2.3 OM-CONFIRMED-ACTION (OM-acción confirmada)

6.2.3.1 Primitiva de servicio

El servicio OM-CONFIRMED-ACTION se muestra en el cuadro 2. La acción específica que ha de realizarse se interpreta en el contexto de la clase de objeto especificada. Este servicio es un servicio confirmado (se envía siempre un informe de éxito o fracaso).

Cuadro 2/Q.754 – Servicio OM-CONFIRMED-ACTION

Nombre de parámetro	pet./ind.	resp./conf.
CallingPartyAddress	M	M
CalledPartyAddress	M	M
DialogueID	M	M
InvokeID	M	M
AccessControl	O	–
BaseManagedObjectClass	M	–
BaseManagedObjectInstance	M	–
ActionInfo	M	–
ActionResult	–	M ^{a)}
ActionError		M ^{b)}
Timer	M ^{c)}	–
a) Obligatorio en el componente Devolver resultado (puede estar vacío). b) Obligatorio en el componente Devolver error. c) Este parámetro se encuentra únicamente en la primitiva de petición.		

Definiciones de parámetro

CallingPartyAddress: Véase el cuadro 1.

CalledPartyAddress: Véase el cuadro 1.

DialogueID: En correspondencia con transaction ID mediante TCAP, como se define en la Recomendación Q.772.

InvokeID: Definida en la Recomendación Q.772.

AccessControl: Información que ha de utilizarse como entrada a funciones de control de acceso.

BaseManagedObjectClass: Identifica la clase de objetos para los cuales se define esta acción.

BaseManagedObjectInstance: Identifica la instancia de objeto sobre la cual ha de realizarse la acción.

ActionInfo: Es una secuencia de ActionType y ActionInfoArg (facultativa). ActionType se define mediante el macro CNF-ACTION y especifica una acción particular que va a realizarse en la instancia de objeto. ActionInfoArg contiene los parámetros para la acción que va a ejecutarse.

ActionResult: Este campo contiene el resultado de la acción realizada con éxito, si procede.

ActionError: Este campo indica información de estado de error o de problema si la acción no se completó con éxito.

Timer: Este parámetro contiene el valor particular del periodo inactivo de espera de respuesta. Toma el valor T₁ para MRVT, T₂ para SRVT o T_c para CVT.

El valor se encuentra en la Recomendación Q.753.

6.2.3.2 Procedimientos para acción confirmada

6.2.3.2.1 Recepción de la petición OM-CONFIRMED-ACTION

El procedimiento `confirmedAction` se inicia al recibirse la primitiva de petición `OM-CONFIRMED-ACTION`. En este caso, el OMPM construye una APDU que solicita la operación `confirmedAction` y transmite la APDU utilizando el servicio `TC-INVOKE` y `TC-BEGIN`.

La primitiva de petición `TC-INVOKE` contiene los siguientes parámetros y valores:

- Operación – Toma el valor de `confirmedAction`.
- Clase – Su valor es 1.
- Parámetros – Corresponde a los parámetros de `confirmedAction` definidos por la palabra clave "PARAMETER" de la definición de operación. El valor "testRoute" se obtiene a partir de `CNF-ACTION` de la `localForm` del `ActionTypeId` procedente de `actionType` de `ActionInfo`.
- Tiempo de inactividad – Se copia del parámetro "Timer" en la petición `OM-CONFIRMED-ACTION`.
- Invoke ID y Dialogue ID se copian de la petición `OM-CONFIRMED-ACTION`.

La primitiva de petición `TC-BEGIN` utiliza los siguientes parámetros y valores:

- Dialogue ID – Como en `TC-INVOKE`.
- Dirección de destino – La `CalledPartyAddress` de la petición `OM-CONFIRMED-ACTION`.
- Dirección de origen – La `CallingPartyAddress` de la petición `OM-CONFIRMED-ACTION`.

La primitiva de petición `N-UNITDATA` emitida a la SCCP tras la recepción de estas primitivas de petición `TC` debe contener el parámetro control de secuencia puesto a "secuencia no garantizada", y el parámetro opción de retorno puesto a "en caso de error retornar mensaje". Véase 2.2.2/Q.711.

6.2.3.2.2 Recepción de TC-BEGIN con indicación TC-INVOKE

En este caso, si la APDU está bien constituida y solicita la operación `confirmedAction`, OMPM emite una primitiva de indicación `OM-CONFIRMED-ACTION` al usuario OMASE.

Si la APDU no está bien constituida, OMPM ignora las indicaciones `TC`.

Todo problema local debe resolverse mediante un mecanismo dependiente de la realización similar al definido en 3.3.4/Q.774.

Si la APDU contiene parámetros adicionales, el OMPM los pasa de forma transparente al usuario OMASE.

6.2.3.2.3 Recepción de la respuesta OM-CONFIRMED-ACTION

La primitiva de respuesta `OM-CONFIRMED-ACTION` puede contener el parámetro `ActionResult` o el parámetro `ActionError`.

El parámetro `ActionResult` indica que la ejecución de la operación ha tenido éxito y el OMPM emite una primitiva de petición `TC-RESULT-L`. Si se trataba de una CVT, se incluyen los siguientes parámetros en `TC-RESULT-L`:

- Operación – Toma el valor de `confirmedAction`.
- Parámetros – Corresponde al parámetro `Success` de `ACTIONRESULT` para la CVT.

La presencia del parámetro ActionError indica que la operación no ha tenido éxito, y el OMPM emite una primitiva de petición TC-U-ERROR con los siguientes parámetros:

- Error – Toma el valor de error adecuado del conjunto definido tras la palabra "ERRORS" de la definición de operación.
- Parámetros – Responde a los parámetros definidos tras la palabra "PARAMETER" de la definición del error.

OMPM transmite el resultado de la operación emitiendo una petición TC-END con los parámetros Dialogue Id y Termination, indicando este último "fin básico".

La primitiva de petición N-UNITDATA emitida en última instancia a la SCCP tras la recepción de estas primitivas de petición TC debe contener el parámetro control de secuencia puesto a "secuencia garantizada", y el parámetro opción de retorno debe indicar "en caso de error descartar mensaje". Véase 2.2.2/Q.711.

6.2.3.2.4 Recepción de TC-END con indicación TC-RESULT-L

En este caso, si la APDU está bien constituida, OMPM emite una primitiva de confirmación OM-CONFIRMED-ACTION con el parámetro ActionResult al usuario OMASE (incluyendo Dialogue ID).

Si la APDU no está bien constituida, OMPM ignora las primitivas TC³.

6.2.3.2.5 Recepción de TC-END con indicación TC-U-ERROR

Si la APDU está bien constituida, OMPM emite una primitiva de confirmación OM-CONFIRMED-ACTION con el parámetro ActionError (y Dialogue ID) al usuario OMASE.

Si la APDU no está bien constituida, OMPM ignora las primitivas TC³.

6.2.3.2.6 Recepción de la indicación TC-L-CANCEL

Esto ocurre si expira el temporizador de invocación.

En este caso, OMPM emite una primitiva de confirmación OM-CONFIRMED-ACTION, con el error específico "fallo" para CNF-ACTION y si la operación invocada fue testRoute, el parámetro failureType indica timerExpired.

El OMPM finaliza el diálogo con una primitiva de petición TC-END, indicando el parámetro Termination "fin previamente acordado".

6.2.3.2.7 Recepción de TC-BEGIN o TC-END con las indicaciones TC-L-REJECT

En la figura 2a se ilustra la recepción de TC-BEGIN con la indicación TC-L-REJECT.

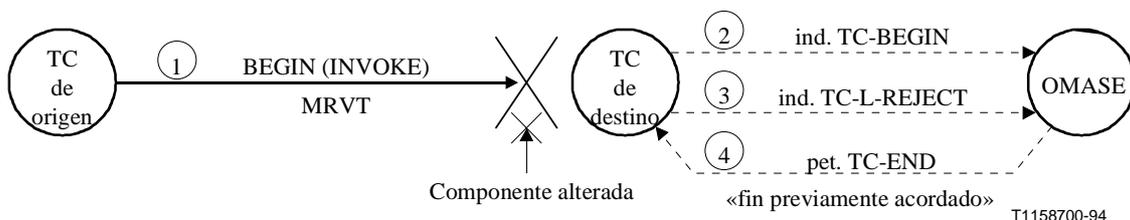


Figura 2a/Q.754

³ La utilización del tiempo de guarda global en el usuario OMASE del nodo iniciador de la prueba permite el fallo armonioso de esta prueba en estas circunstancias.

Si OMPM recibe una indicación TC-L-REJECT con una indicación TC-BEGIN, finaliza el diálogo emitiendo una primitiva de petición TC-END indicando el parámetro Terminación "fin previamente acordado".

Si OMPM recibe una indicación TC-L-REJECT con una indicación TC-END, emite una primitiva de confirmación OM-CONFIRMED-ACTION con el error específico "fallo" de CNF-ACTION y, si se invocó testRoute, el parámetro failureType de la primitiva de confirmación indica processingFailure ("fallo de procesamiento"). Esto se ilustra en la figura 2b.

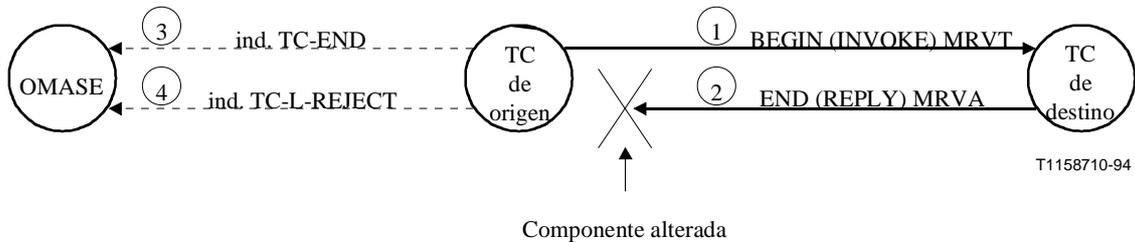


Figura 2b/Q.754

6.2.3.2.8 Recepción de TC-END con la indicación TC-R-REJECT

En este caso, OMPM emite una primitiva de confirmación OM-CONFIRMED-ACTION con el error específico "fallo" de CNF-ACTION y, si se invocó TestRoute, el parámetro failureType de la primitiva de confirmación indica processing Failure ("fallo de procesamiento").

6.2.3.2.9 Recepción de la indicación TC-P-ABORT

Esto se ilustra en los dos diagramas de la figura 2c.

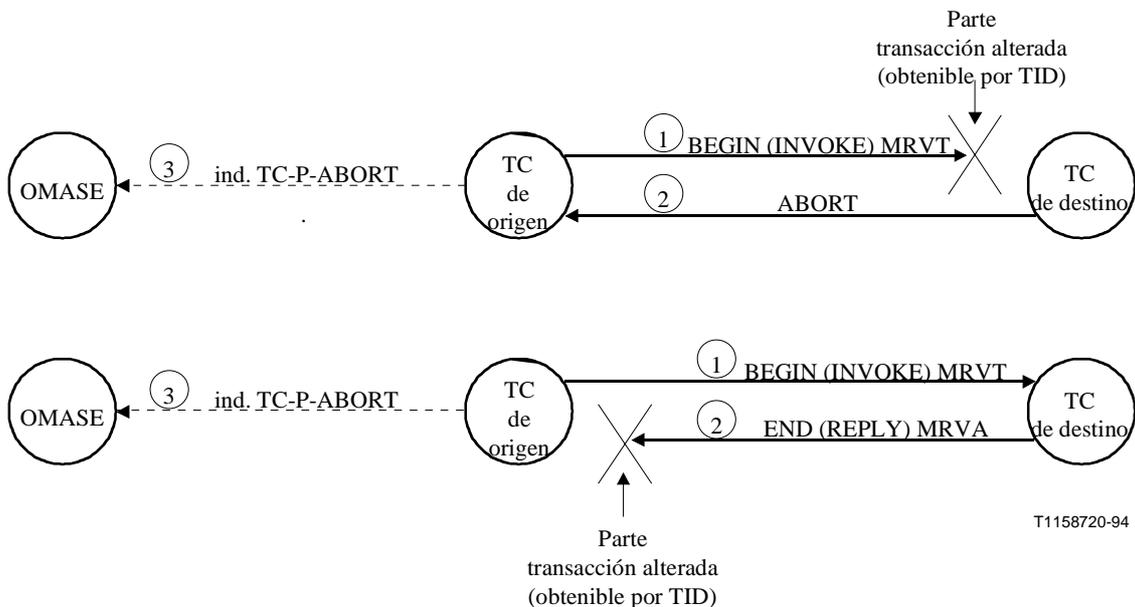


Figura 2c/Q.754

En este caso, OMPM emite una primitiva de confirmación OM-CONFIRMED-ACTION con el error específico "fallo" de CNF-ACTION y, si se invocó testRoute, el parámetro failureType de la primitiva de confirmación indica processingFailure ("fallo de procesamiento").

6.2.3.2.10 Recepción de TC-NOTICE

En este caso, OMPM emite una primitiva de confirmación OM-CONFIRMED-ACTION con el error específico "fallo" de CNF-ACTION y, si se invocó testRoute, el parámetro failureType de la primitiva de confirmación indica processingFailure ("fallo de procesamiento").

Definición de los errores

En la definición de los dos servicios OM se hace referencia a un cierto número de errores; los mismos se definen a continuación.

Definiciones

noSuchObjectClass: La clase de objeto de la APDU invocación no es reconocida por el extremo de recepción.

noSuchObjectInstance: Si bien la clase de objeto de la APDU invocación se reconoce, no hay ninguna instancia de objeto correspondiente de dicha clase en el extremo de recepción.

accessDenied: No se permite el acceso al recurso.

processingFailure: Aparece un fallo en el procesamiento de una acción o suceso específico. Los indicadores de fallo son específicos de la acción o el suceso.

noSuchAction: El tipo de acción no es soportado o conocido por el extremo de recepción.

noSuchArgument: El argumento especificado no es conocido o soportado por el extremo de recepción.

invalidArgumentValue: El valor del argumento no es adecuado para el extremo de recepción.

6.3 Sintaxis abstracta del protocolo OMASE

Véase la figura 3.

```

-- OMASE protocol --
OMASE { itu-t(0) recommendation q 754 omase(0) version2(2) }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN
-- TCAP definitions --
EXPORTS EVERYTHING;

-- the OPERATION and ERROR information objects defined here are equivalent to the respective MACROs in
-- TCAPMessages {ccitt recommendation q 773 modules(2) messages(1) version2(2) } of Rec. Q.773 (1993) --

OPERATION ::= CLASS
{
    &ArgumentType      OPTIONAL,
    &ResultType        OPTIONAL,
    &Errors             ERROR OPTIONAL,
    &Linked             OPERATION OPTIONAL,
    &operationCode     Code UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [PARAMETER &ArgumentType]
    [RESULT &ResultType]
    [ERRORS &Errors]
    [LINKED &Linked]
    [CODE &operationCode]
}

ERROR ::= CLASS
{
    &ParameterType    OPTIONAL,
    &errorCode         Code UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [PARAMETER &ParameterType]
    [CODE &errorCode]
}

Code ::= CHOICE
{
    localValue        INTEGER.
    GlobalValue       OBJECT IDENTIFIER
}

```

Figura 3/Q.754 (hoja 1 de 11) – Sintaxis formal de los servicios OMASE

-- OMASE operators --

```
eventReport OPERATION ::=
{
PARAMETER eventReportArgument      EventReportArgument

CODE      localValue:0
}

confirmedAction OPERATION ::=
{
PARAMETER actionArgument      ActionArgument
RESULT   actionResult      ActionResult
ERRORS   { accessDenied | invalidArgumentValue |
          noSuchAction | noSuchArgument |
          noSuchObjectClass | noSuchObjectInstance |
          processingFailure }

CODE     localValue:7
}
```

Figura 3/Q.754 (hoja 2 de 11) – Sintaxis formal de los servicios OMASE

-- The om-service error definitions are as follows:

```
noSuchObjectClass      ERROR ::=
  {
    PARAMETER      ObjectClass
    CODE           localValue:0
  }

noSuchObjectInstance  ERROR ::=
  {
    PARAMETER      ObjectInstance
    CODE           localValue:1
  }

accessDenied          ERROR ::=
  {
    CODE           localValue:2
  }

noSuchAction          ERROR ::=
  {
    PARAMETER      NoSuchAction
    CODE           localValue:9
  }

processingFailure     ERROR ::=
  {
    PARAMETER      ProcessingFailure -- optional --
    CODE           localValue:10
  }

noSuchArgument        ERROR ::=
  {
    PARAMETER      NoSuchArgument
    CODE           localValue:14
  }

invalidArgumentValue  ERROR ::=
  {
    PARAMETER      InvalidArgumentValue
    CODE           localValue:15
  }
```

Figura 3/Q.754 (hoja 3 de 11) – Sintaxis formal de los servicios OMASE

-- The following gives the supporting type definitions: --

```

ActionArgument ::= SEQUENCE {
    COMPONENTS OF
    accessControl
    actionInfo
    BaseManagedObjectId,
    [5] AccessControl OPTIONAL,
    [12] IMPLICIT ActionInfo }

ActionInfo ::= SEQUENCE {
    actionType
    actionInfoArg
    [3] IMPLICIT CNF-ACTION.&operationCode,
    [4] CNF-ACTION. &ActionArgType(@actionType)
    OPTIONAL}

ActionResult ::= SEQUENCE {
    managedObjectClass
    managedObjectInstance
    currentTime
    actionReply
    ObjectClass OPTIONAL,
    ObjectInstance OPTIONAL,
    [5] IMPLICIT GeneralizedTime OPTIONAL,
    [6] IMPLICIT ActionReply OPTIONAL }

ActionTypeId ::= CHOICE {
    -- globalForm... --
    localForm
    [3] IMPLICIT CNF-ACTION }

BaseManagedObjectId ::= SEQUENCE {
    baseManagedObjectClass
    baseManagedObjectInstance
    ObjectClass,
    ObjectInstance }

EventReportArgument ::= SEQUENCE {
    managedObjectClass
    managedObjectInstance
    eventTime
    eventType
    eventInfo
    ObjectClass,
    ObjectInstance,
    [5] IMPLICIT GeneralizedTime OPTIONAL,
    [7] IMPLICIT EVENT. &operationCode
    [8] EVENT. &EventInfo Type (@eventType)
    OPTIONAL}

EventTypeId ::= CHOICE {
    -- globalForm... --
    localForm
    [7] IMPLICIT EVENT }

ActionReply ::= SEQUENCE {
    actionType
    actionReplyInfo
    [3] IMPLICIT CNF-ACTION.
    &operationCode,
    [4]
    CNF-ACTION.&ActionResultType(@actionType)
    }

AccessControl ::= EXTERNAL
-- AccessControl syntax is to be compatible with that defined in CMIP coded X.209 --

```

Figura 3/Q.754 (hoja 4 de 11) – Sintaxis formal de los servicios OMASE

InvalidArgumentValue	::= CHOICE {	actionValue	[0] IMPLICIT ActionInfo,
		eventValue	[1] IMPLICIT SEQUENCE {
		eventType	[7] IMPLICIT EVENT.&operationCode,
		eventInfo	[8] EVENT.&EventInfoType(@eventType)
			OPTIONAL }
NoSuchAction	::= SEQUENCE {	managedObjectClass	ObjectClass,
		actionType	ActionTypeId }
NoSuchArgument	::= CHOICE {	actionId	[0] IMPLICIT SEQUENCE {
		managedObjectClass	ObjectClass OPTIONAL,
		actionType	ActionTypeId },
		eventId	[1] IMPLICIT SEQUENCE {
		managedObjectClass	ObjectClass OPTIONAL,
		eventType	EventTypeID }
ObjectClass	::= CHOICE {	globalForm	[0] IMPLICIT OBJECT
		-- ... --	IDENTIFIER,
		}	
ObjectInstance	::= CHOICE {	-- ... --	
		nonSpecificForm	[3] IMPLICIT OCTET STRING,
		-- ... --	
		}	
ProcessingFailure	::= SEQUENCE {	managedObjectClass	ObjectClass OPTIONAL,
		managedObjectInstance	ObjectInstance OPTIONAL,
		specificErrorInfo	[5] IMPLICIT SpecificErrorInfo }
SpecificError	::= INTEGER	-- defined by object class --	
SpecificErrorInfo	::= SEQUENCE {	errorType	[0] IMPLICIT SpecificError.&errorCode,
		errorParm	[1] IMPLICIT SPECIFIC-ERROR. &ProcessingError
			ParmType(@errorType) OPTIONAL }
Timer	::= INTEGER	-- seconds --	

Figura 3/Q.754 (hoja 5 de 11) – Sintaxis formal de los servicios OMASE

-- Specific event reports are categorized by object class. The protocol uses may be described
-- by the EVENT MACRO below. --

```
EVENT ::= CLASS
{
    &EventInfoType          OPTIONAL,
    &operationCode          INTEGER UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [EVENTINFO              &EventInfoType]
    [CODE                   &operationCode]
}
```

-- Specific Actions are categorized by object class. The protocol uses may be described
-- by the CNF-ACTION INFORMATION OBJECT below. --

```
CNF-ACTION ::= CLASS
{
    &ActionArgType          OPTIONAL,
    &ActionResultType      OPTIONAL,
    &SpecificErrors         SPECIFIC-ERROR OPTIONAL,
    &operationCode          INTEGER UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [ACTIONARG              &ActionArgType]
    [ACTIONRESULT          &ActionResultType]
    [SPECIFICERRORS        &SpecificErrors]
    [CODE                   &operationCode]
}
```

-- Errors that are action or event specific are defined using the SPECIFIC-ERROR macro below. --

```
SPECIFIC-ERROR ::= CLASS
{
    &ProcessingErrorParmType OPTIONAL,
    &errorCode               INTEGER UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [PARAMETER              &ProcessingErrorParmType]
    [CODE                   &errorCode]
}
```

Figura 3/Q.754 (hoja 6 de 11) – Sintaxis formal de los servicios OMASE

-- specific OMASE constructs follow --

```
testRoute CNF-ACTION ::=
{
    ACTIONARG SEQUENCE{
        initiating SP          [0] IMPLICIT PointCode,
        traceRequested        [1] IMPLICIT BOOLEAN,
        threshold              [2] IMPLICIT INTEGER,
        pointCodesTraversed   [3] IMPLICIT PointCodeList,
        formindicator         [4] IMPLICIT Formindicator OPTIONAL,
-- formIndicator is required in SRVT, but not used in MRVT --

        mtpBackwardRoutingRequested [5] IMPLICIT BOOLEAN OPTIONAL,
-- mtpBackwardRoutingRequested is required in SRVT, but not in MRTV --

        testInitiatorGT       [6] IMPLICIT GlobalTitle OPTIONAL,
        destinationPC         [7] IMPLICIT PointCode OPTIONAL,
        destinationSSN        [8] IMPLICIT SubsystemNumber
        OPTIONAL,
        backupDPC             [9] IMPLICIT PointCode OPTIONAL,
        backupSSN             [10] IMPLICIT SubsystemNumber
        OPTIONAL,
        originalGT            [11] IMPLICIT GlobalTitle OPTIONAL,
        inputGT               [16] IMPLICIT GlobalTitle OPTIONAL,
-- parameters with tags 4 through 12 can only be used in SRVT, not MRVT --

        routePriorityList     [12] IMPLICIT RoutePriorityList
        OPTIONAL,
-- routePriorityList can only be used in MRVT, and only if the infoRequest parameter is present. --

        infoRequest           [13] IMPLICIT BIT STRING {
            pointCode(0),
            pointCodeList(1),
            routePriorityList(2),
            ...} OPTIONAL,
-- infoRequest is used to indicate that the test initiator node can accept a route TraceNew
-- RVR message, and also asks for particular parameters to be returned in it, if it is sent. This
-- parameter can only be inserted at the initiator node, but it can be copied into regenerated MRVTs. --

        returnUnknownParams   [14] IMPLICIT BIT STRING {
            tag15(0),
            tag16(1),
            ...} OPTIONAL,
-- returnUnknownParams is used to indicate which parameters that a node does not understand
-- should be returned in an RVR if one is sent (or in an RVA message in the copyData field
-- if the test initiator is unknown). Bit 0 represents an RVT parameter with tag value 15, bit 1
-- an RVT parameter with tag value 16, etc. To avoid confusion in the copyData field, when
-- defining a new parameter in the RVR message, the tag should have the same value as it has
-- in the RVT message. This parameter can only be present if infoRequest is present. --

        directRouteCheck     [15] IMPLICIT BOOLEAN OPTIONAL,
-- directRouteCheck can only be used in MRVT. --

        ... }

SPECIFICERRORS          { failure | partialSuccess }
CODE                    1
}

-- TC timer = T1 for MRVT, = T2 for SRVT, Class = 1--
```

Figura 3/Q.754 (hoja 7 de 11) – Sintaxis formal de los servicios OMASE

PointCode ::= OCTET STRING
PointCodeList ::= SEQUENCE OF PointCode
RoutePriorityList ::= SEQUENCE OF Priority

Priority ::= INTEGER{
 unknown(0),
 firstChoice(1),
 secondChoice(2),
 thirdChoice(3),
 ...} (0..255)

FormIndicator ::= INTEGER
 { compare (0),
 noCompare (1) } (0..1)

GlobalTitle ::= OCTET STRING
 -- the GlobalTitle here consists of the SCCP GTI + GT, the GTI should be encoded exactly as in
 -- 3.4.1/Q.713, and the GT as in 3.4.2.1 to 3.4.2.4/Q.713 as appropriate. --

SubsystemNumber ::= OCTET STRING

failure SPECIFIC-ERROR ::=
 {
 PARAMETER SEQUENCE **{failureType** **[0] IMPLICIT FailureString,**
 traceSent **[1] IMPLICIT BOOLEAN,**
 copyData **[2] IMPLICIT CopyData OPTIONAL,**
 -- copyData might be present if failureType is unknownInitiatingSp, traceSent is FALSE,
 -- and the prompting RVT message contained a requestInfo parameter, or returnUnknownParams
 -- was in the RVT message. --
 ... }
 CODE **1**
 }

FailureString ::= BIT STRING
 { detectedLoop(0),
 excessiveLengthRoute(1),
 unknownDestination(2),
 routeInaccessible(3),
 processingFailure(4),
 unknownInitiatingSP(5),
 timerExpired(6),
 sPNotAnSTP(7),
 -- wrongSp is a synonym, used in SRVT, of sPNotAnSTP. --
 incorrectTranslation-Primary (8),
 incorrectTranslation-Secondary (9),
 incorrectTranslation-Intermediate (10),
 notPrimaryDestination (11),
 notSecondaryDestination (12),
 notRecognizedPrimary (13),
 notRecognizedSecondary (14),
 routingProblem (15),
 -- bits 8 through 15 might only be set in SRVT, not MRVT. --
 maxNrMRVTestsAlready(16),
 -- maxNrSRVTestsAlready is a synonym, used in SRVT, of maxNrMRVTestsAlready. --
 indirectRoute(17),
 -- indirectRoute might only be set in MRVT, not SRVT. --
 ... }

CopyData ::= OCTET STRING

Figura 3/Q.754 (hoja 8 de 11) – Sintaxis formal de los servicios OMASE

```

partialSuccess    SPECIFIC-ERROR ::=
  {
    PARAMETER SEQUENCE    {failureType    [0] IMPLICIT FailureString,
                             traceSent      [1] IMPLICIT BOOLEAN,
                             copyData       [4] IMPLICIT CopyData OPTIONAL,

    -- copyData might be present if failureType is unknownInitiatingSP, traceSent is FALSE,
    -- and the prompting RVT message contained a requestInfo parameter,
    -- or returnUnknownParams was in the RVT message. --

    ... }
    CODE      2
  }

routeTrace      EVENT ::=
  {
    EVENTINFO CHOICE {
      success          [0] IMPLICIT PointCodeList,
      detectedLoop     [1] IMPLICIT PointCodeList,
      excessiveLengthRoute [2] IMPLICIT PointCodeList,
      unknownDestination [3] IMPLICIT NULL,
      routeInaccessible [4] IMPLICIT PointCode,
      processingFailure  [5] IMPLICIT NULL,
      unknownInitiatingSP [6] IMPLICIT PointCode,
      timerExpired      [7] IMPLICIT PointCodeList,
      sPNotAnSTP        [8] IMPLICIT PointCodeList,

      -- wrongSP is a synonym, used in SRVT, for sPNotAnSTP. --

      incorrectTranslation-Primary [9] IMPLICIT PointCodeList,
      incorrectTranslation-Secondary [10] IMPLICIT PointCodeList,
      incorrectTranslation-Intermediate [11] IMPLICIT PointCodeList,
      notPrimaryDestination [12] IMPLICIT PointCodeList,
      notSecondaryDestination [13] IMPLICIT PointCodeList,
      notRecognizedPrimary [14] IMPLICIT PointCodeList,
      notRecognizedSecondary [15] IMPLICIT PointCodeList,
      routingProblem [16] IMPLICIT PointCodeList

      -- the choices with tags 9 through 16 can only be used in SRVT. --

    }
    CODE      2
  }

  -- TC Timer = 0, Class = 4

```

Figura 3/Q.754 (hoja 9 de 11) – Sintaxis formal de los servicios OMASE

```

routeTraceNew EVENT ::=
{
    EVENTINFO SEQUENCE {
        result [0] IMPLICIT ErrorTag,
        pointCode [1] IMPLICIT PointCode OPTIONAL,
        pointCodeList [2] IMPLICIT PointCodeList OPTIONAL,
        routePriorityList [3] IMPLICIT RoutePriorityList OPTIONAL,
        copyData [4] IMPLICIT CopyData OPTIONAL,
        -- copyData allows any parameters included in an RVA message, when the
        -- test initiator is unknown, to be copied into the RVR, without enhancing it. It also
        -- allows new OPTIONAL RVT parameters not understood by the node generating the
        -- RVR from the RVA message to be returned, when requested by the test initiator.
        -- Note that a new parameter defined in routeTraceNew should, if it is also defined in
        -- testRoute, have the same tag value as in testRoute.
        -- One RVR message should be sent for each error detected (no error diagnostics should be
        -- "or'd" together). --
        ... }

    CODE 4
}
-- TC Timer = 0, Class = 4

```

Figura 3/Q.754 (hoja 10 de 11) – Sintaxis formal de los servicios OMASE

```

ErrorTag ::=INTEGER {
    success(0),
    detectedLoop(1),
    excessiveLengthRoute(2),
    unknownDestination(3),
    routeInaccessible(4)
    processingFailure(5)
    unknownInitiatingS(6),
    timerExpired(7),

    -- wrongSP is a synonym, used in SRVT, of sPNotAnSTP. --

    incorrectTranslation-Primary(9),
    incorrectTranslation-Secondary(10),
    incorrectTranslation-Intermediate(11),
    notPrimaryDestination(12),
    notSecondaryDestination(13),
    notRecognizedPrimary(14),
    notRecognizedSecondary(15),
    routingProblem(16)
    -- values 9 through 16 are applicable only in SRVT, not in MRVT. --

    maxNrMRVTestsAlready(17),
    -- maxNrSRVTestsAlready is a synonym, used in SRVT, of maxNrMRVTestsAlready. --

    indirectRoute(18),
    -- value 18 is applicable only in MRVT, not SRVT. --

    ... } (0..255)

END -- OMASE protocol --

```

Figura 3/Q.754 (hoja 11 de 11) – Sintaxis formal de los servicios OMASE

ANEXO A

Utilización de las interfaces de las primitivas

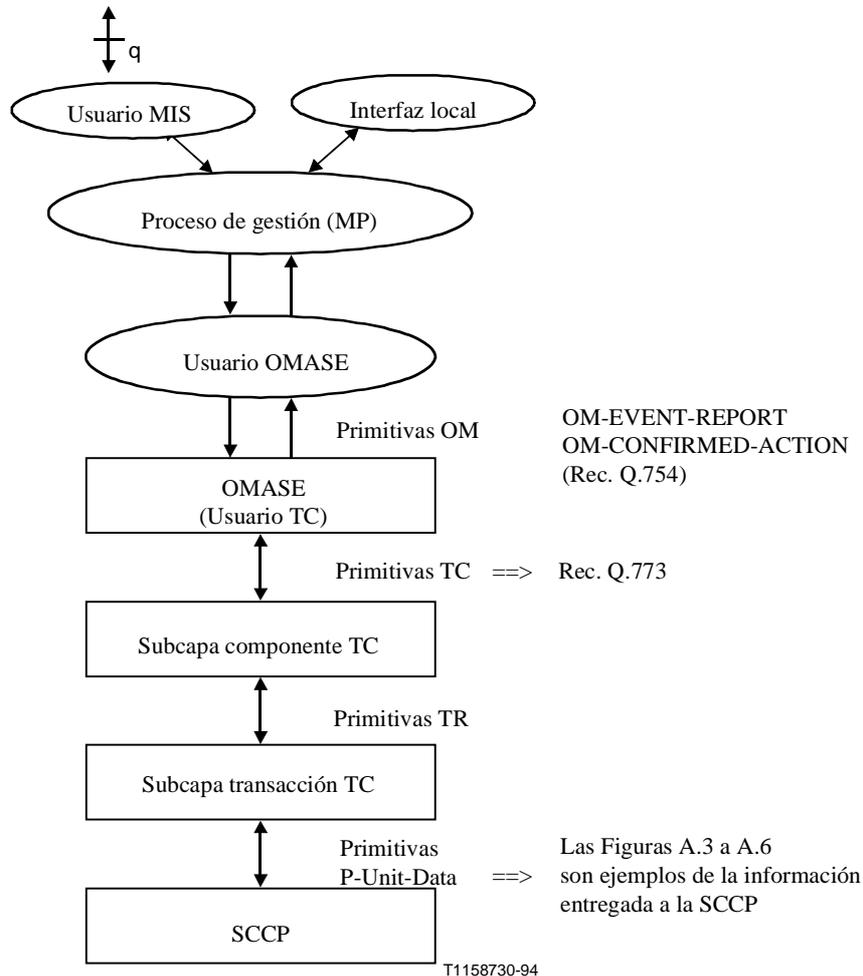


Figura A.1/Q.754 – Interfaces de las primitivas

La siguiente figura A.2 ilustra la utilización de las primitivas en una prueba MRV. El usuario OMASE del origen, al recibir una petición "sendMRVT" del proceso de gestión (MP – véase la Recomendación Q.753 sobre el modelo utilizado), construye una petición OM-CONFIRMED-ACTION. La secuencia es la mostrada por la secuencia de primitivas y mensajes, hasta el número 5. En ese punto, si el nodo no es el destino probado, el usuario OMASE que recibe la indicación OM-CONFIRMED-ACTION solicita a OM-CONFIRMED-ACTION de OMASE que envíe mensajes MRVT por todas las rutas al destino probado que figuran en el cuadro de encaminamiento. Cuando se reciben todos los mensajes MRVA (considerados por el usuario OMASE como primitivas de confirmación OM-CONFIRMED-ACTION), el usuario OMASE emite la primitiva de respuesta OM-CONFIRMED-ACTION como se indica en 6.

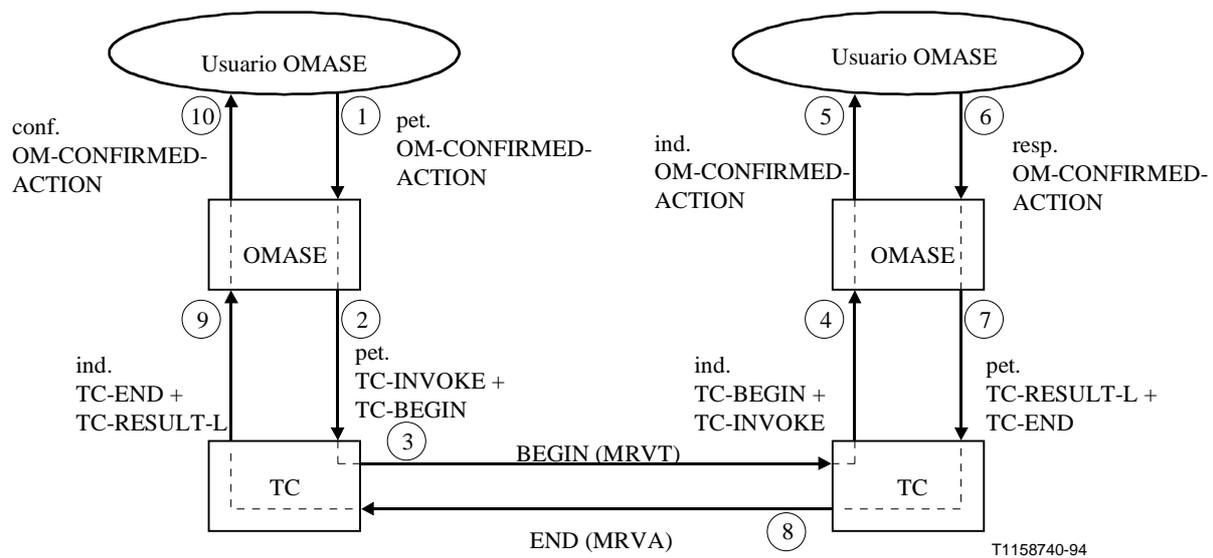


Figura A.2/Q.754 – Ejemplo de utilización de las interfaces de las primitivas

Nombre del campo	Codificación de los bits	Referencia/explicación
Rótulo de tipo de mensaje	01100010	= Comenzar (BEGIN) (cuadro 8/Q.773)
Longitud de mensaje	00110010	50 octetos que siguen a la parte TC
Rótulo de ID de transacción	01001000	= Originador (cuadro 10/Q.773)
Longitud	00000100	4 octetos
Valor de ID de transacción	xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx	TCAP basado en un diálogo a nivel de usuario
Rótulo de porción de componente	01101100	(cuadro 14/Q.773)
Longitud	00101010	Los 42 octetos que siguen aquí
Rótulo de tipo de componente	10100001	= Invocar (cuadro 19/Q.773)
Longitud	00101000	Los 40 octetos que siguen aquí
Rótulo de ID de componente	00000010	= ID de invocar (cuadro 20/Q.773)
Longitud	00000001	1 octeto
Valor de ID de invocar	xxxxxxxx	PROPORCIONADO por el OMAP
Rótulo de código de operación	00000010	= Local (cuadro 22/Q.773)
Longitud	00000001	1 octeto
Código de operación	00000111	= Acción confirmada (Figura 3/Q.754)
Rótulo de secuencia de parámetro	00110000	= Rótulo de secuencia de parámetro (cuadro 23/Q.773)
Longitud	00100000	Los 32 octetos que siguen aquí
Rótulo de clase de objeto	10000000	globalForm X.711 y X.690
Longitud	00000101	5 octetos
Valores – Tablas de encaminamiento por la MTP	00000000	Rec. UIT-T
	00010001	q
	10000101	85 => 754
	01110010	72 =>
	00000000	Tablas encaminamiento por la MTP 1992
Rótulo de instancia de objeto	10000011	nonSpecificForm X.711 y X.690
Longitud	00000010	2 octetos
Valor de instancia de objeto	xxxxxxxx xxxxxxxx	Destino probado (OMAP)
Rótulo de información de acción	10101100	Recs. X.711 y X.690
Longitud	00010011	Los 19 octetos que siguen aquí
Rótulo de tipo de acción	10000011	Forma local Recs. X.711 y X.690
Longitud	00000001	1 octeto
CNF-ACTION	00000001	= testRoute (figura 3/Q.754)
Rótulo de Arg de info de acción	10100100	Recs. X.711 y X.690
Longitud	00001110	Los 14 octetos que siguen aquí
Rótulo de secuencia de parámetro	00110000	= Rótulo de secuencia (cuadro 23/Q.773)
Longitud	00001100	Los 12 octetos que siguen aquí
Rótulo de SP iniciador	10000000	Figura 3/Q.754, Rec. X.690
Longitud	00000010	2 octetos
Valor de SP iniciador	xxxxxxxx xxxxxxxx	(OMAP) iniciador de prueba
Rótulo de petición de rastreo	10000001	Figura 3/Q.754, Rec. X.690
Longitud	00000001	1 octeto
Valor	00000001	= TRUE
Rótulo de umbral	10000010	= umbral (figura 3/Q.754)
Longitud	00000001	1 octeto
Valor de umbral	xxxxxxxx	PROPORCIONADO por el OMAP
Rótulo de código de puntos atravesados	10100011	Figura 3/Q.754
Longitud	00000000	Lista vacía de códigos de punto

Figura A.3/Q.754 – Ejemplo de un mensaje MRVT entregado a la SCCP

Nombre del campo	Codificación de los bits	Referencia/explicación
Rótulo de tipo de mensaje	01100100	= FIN (END) (cuadro 8/Q.773)
Longitud de mensaje	00001101	13 octetos que siguen a la parte TC
Rótulo de ID de transacción	01001001	= Destino (cuadro 10/Q.773)
Longitud	00000100	4 octetos
Valor de ID de transacción	xxxxxxx xxxxxxx xxxxxxx xxxxxxx	Igual que en COMENZAR (BEGIN) (mensaje MRVT)
Rótulo de porción de componente	01101100	(cuadro 14/Q.773)
Longitud	00000101	Los 5 octetos que siguen aquí
Rótulo de tipo de componente	10100010	= Ret. Res (L) (cuadro 19/Q.773)
Longitud	00000011	Los 3 octetos que siguen aquí
Rótulo de ID de componente	00000010	= ID de invocar (cuadro 20/Q.773)
Longitud	00000001	1 octeto
Valor de ID de invocar	xxxxxxx	Igual que el mensaje MRVT (correlación)

Figura A.4/Q.754 – Ejemplo de un mensaje MRVA (éxito) entregado a la SCCP

Nombre del campo	Codificación de los bits	Referencia/explicación
Rótulo de tipo de mensaje	01100100	= Fin (END) (cuadro 8/Q.773)
Longitud de mensaje	00100000	32 octetos que siguen a la parte TC
Rótulo de ID de transacción	01001001	= Destino (cuadro 10/Q.773)
Longitud	00000100	4 octetos
Valor de ID de transacción	xxxxxxx xxxxxxx xxxxxxx	Igual que en comienzo (BEGIN) (mensaje MRVT)
Rótulo de porción de componente	01101100	(cuadro 14/Q.773)
Longitud	00011000	Los 24 octetos que siguen aquí
Rótulo de tipo de componente	10100011	= Retornar error (cuadro 19/Q.773)
Longitud	00010110	Los 22 octetos que siguen aquí
Rótulo de ID de componente	00000010	= ID de invocar (cuadro 20/Q.773)
Longitud	00000001	1 octeto
Valor de ID de invocar	xxxxxxx	Igual que para mensaje MRVT (correlación)
Rótulo de código de error	00000010	cuadro 24/Q.773 (local)
Longitud	00000001	1 octeto
Fallo de procesamiento	00001010	Figura 3/Q.754
Rótulo de secuencia de parámetro	00110000	= Rótulo de secuencia (cuadro 23/Q.773)
Longitud	00001110	Los 14 octetos que siguen aquí
Rótulo info de error específico	10100101	Figura 3/Q.754
Longitud	00001100	Los 12 octetos que siguen aquí
Rótulo de tipo de error	10000000	Figura 3/Q.754
Longitud	00000001	1 octeto
Fallo	00000001	Figura 3/Q.754
Parámetros de error	10100001	Figura 3/Q.754
Longitud	00000111	Los 7 octetos que siguen aquí
Rótulo de tipo de fallo	10000000	Figura 3/Q.754
Longitud	00000010	2 octetos
Bits no utilizados	00000000	Ningún bit
Cadena de fallo	xxxxxxx	Depende del tipo de fallo (figura 3/Q.754)
Rótulo de rastreo enviado	10000001	Figura 3/Q.754
Longitud	00000001	1 octeto
Valor de rastreo enviado	0000000x	True = 1, False = 0, (figura 3/Q.754)

Figura A.5/Q.754 – Ejemplo de un mensaje MRVA (fallo) entregado a la SCCP

Nombre del campo	Codificación de los bits	Referencia/explicación
Rótulo de tipo de mensaje	01100010	= comenzar (BEGIN) (cuadro 8/Q.773)
Longitud de mensaje	00101100	44 octetos que siguen a la parte TC
Rótulo de ID de transacción	01001000	= Origen (cuadro 10/Q.773)
Longitud	00000100	4 octetos
Valor de ID de transacción	xxxxxxx xxxxxxx xxxxxxx xxxxxxx	TCAP basado en un diálogo a nivel de usuario
Rótulo de porción de componente	01101100	(cuadro 14/Q.773)
Longitud	00100100	Los 36 octetos que siguen aquí
Rótulo de tipo de componente	10100001	= Invocar (cuadro 19/Q.773)
Longitud	00100010	Los 34 octetos que siguen aquí
Rótulo de ID de componente	00000010	= ID de invocar (cuadro 20/Q.773)
Longitud	00000001	1 octeto
Valor de ID de invocar	xxxxxxx	PROPORCIONADO por la OMAP
Rótulo de código de operación	00000010	= Local (cuadro 22/Q.773)
Longitud	00000001	1 octeto
Código de operación	00000000	= Informe de suceso (figura 3/Q.754)
Rótulo de secuencia de parámetro	00110000	= Rótulo de secuencia (cuadro 23/Q.773)
Longitud	00011010	Los 26 octetos que siguen aquí
Rótulo de clase de objeto	10000000	(figura 3/Q.754)
Longitud	00000101	5 octetos
Valores – Tablas de encaminamiento por la MTP	00000000 00010001 10000101 01110010	Rec. UIT-T q 85 => 754 72 =>
Rótulo de instancia de objeto	00000000	Tablas de encaminamiento por la MTP 1992
Longitud	10000011	(figura 3/Q.754)
Valor de instancia de objeto	00000010 xxxxxxx xxxxxxx	2 octetos PC de terminación (OMAP) <Destino probado>
Rótulo de tipo de suceso	10000111	Figura 3/Q.754
Longitud	00000001	1 octeto
Tipo de suceso	00000010	= routeTrace (figura 3/Q.754)
Rótulo de tipo info de suceso	10101000	Figura 3/Q.754
Longitud	00001010	Los 10 octetos que siguen aquí
Identificador de éxito	10100000	Figura 3/Q.754
Longitud	00001000	Los 8 octetos que siguen aquí
Rótulo de código de punto	00000100	= OCTET STRING
Longitud	00000010	2 octetos
Código de punto	xxxxxxx xxxxxxx	
Rótulo de código de punto	00000100	= OCTET STRING
Longitud	00000010	2 octetos
Código de punto	xxxxxxx xxxxxxx	

Figura A.6/Q.754 – Ejemplo de un mensaje MRVR (éxito) entregado a la SCCP

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Z	Lenguajes de programación