



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

**Q.754**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

(03/93)

**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»)

---

## PREFACIO

El Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) es un órgano permanente de la Unión Internacional de Telecomunicaciones. El UIT-T tiene a su cargo el estudio de las cuestiones técnicas, de explotación y de tarificación y la formulación de Recomendaciones al respecto con objeto de normalizar las telecomunicaciones sobre una base mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se reúne cada cuatro años, establece los temas que habrán de abordar las Comisiones de Estudio del UIT-T, que preparan luego Recomendaciones sobre esos temas.

La Recomendación UIT-T Q.754, preparada por la Comisión de Estudio XI (1988-1993) del UIT-T, fue aprobada por la CMNT (Helsinki, 1-12 de marzo de 1993).

---

## NOTAS

1 Como consecuencia del proceso de reforma de la Unión Internacional de Telecomunicaciones (UIT), el CCITT dejó de existir el 28 de febrero de 1993. En su lugar se creó el 1 de marzo de 1993 el Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T). Igualmente en este proceso de reforma, la IFRB y el CCIR han sido sustituidos por el Sector de Radiocomunicaciones.

Para no retrasar la publicación de la presente Recomendación, no se han modificado en el texto las referencias que contienen los acrónimos «CCITT», «CCIR» o «IFRB» o el nombre de sus órganos correspondientes, como la Asamblea Plenaria, la Secretaría, etc. Las ediciones futuras en la presente Recomendación contendrán la terminología adecuada en relación con la nueva estructura de la UIT.

2 Por razones de concisión, el término «Administración» se utiliza en la presente Recomendación para designar a una administración de telecomunicaciones y a una empresa de explotación reconocida.

© UIT 1994

Reservados todos los derechos. No podrá reproducirse o utilizarse la presente Recomendación ni parte de la misma de cualquier forma ni por cualquier procedimiento, electrónico o mecánico, comprendidas la fotocopia y la grabación en micropelícula, sin autorización escrita de la UIT.

## ÍNDICE

	<i>Página</i>
1	Introducción ..... 1
2	MTP ..... 1
2.1	Prueba de verificación del encaminamiento MTP (MRVT) ..... 1
3	Parte control de la conexión de señalización (SCCP) ..... 9
3.1	ASE de la prueba de verificación del encaminamiento SCCP (SRVT) ..... 9
4	Gestión de circuitos ..... 22
4.1	Elemento de servicio de aplicación de la prueba de validación de circuitos ..... 22
5	Capacidades de transacción (TC) ..... 24
6	Definiciones generales ..... 24
6.1	Objetos y operaciones ..... 24
6.2	Primitivas y procedimientos del protocolo OMASE ..... 25
6.3	Sintaxis abstracta del protocolo OMASE ..... 31
Anexo A	..... 36



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

(Helsinki, 1993)

### 1 Introducción

Debe señalarse que en caso de discrepancia entre las Recomendaciones Q.753 y Q.754, la primera 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*)<sup>1</sup>.

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, *circuits 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.

### 2 MTP

#### 2.1 Prueba de verificación del encaminamiento MTP (MRVT)<sup>2</sup>

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.

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.

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

<sup>1</sup>) CMIP se define en ISO IS 9596 y en la Recomendación X.711.

<sup>2</sup>) Para la descripción de la notación formal véanse las Recomendaciones X.208 y X.209.

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
ActionArg		O/M	Referencias
initiatingSP		M	2.1.1.1.1
traceRequested		M	2.1.1.1.2
threshold		M	2.1.1.1.3
pointCodesTraversed		M	2.1.1.1.4
ActionResult			
vacío			
Linked Operations			
N/A			
Specific Errors			Referencias
failure			2.1.1.3.1
partialSuccess			2.1.1.3.2
O Facultativo			
M Obligatorio			

testRoute CNF-ACTION			
ACTIONINFOARG SEQUENCE {			
	initiatingSP	[0] IMPLICIT PointCode,	
	traceRequested	[1] IMPLICIT BOOLEAN,	
	threshold	[2] IMPLICIT INTEGER,	
	pointCodesTraversed	[3] IMPLICIT PointCodeList	
}			
SPECIFICERRORS		{failure, partialSuccess}	
::= 1			

### 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.	

PointCode ::= OCTET STRING

### 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) 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
<i>Contenido</i>	
Número entero	

### 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. Esta pointCodeList podrá estar vacía.

<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	

PointCodeList ::= SEQUENCE OF PointCode

### 2.1.1.2 Action Results (resultados de acciones)

No hay contenido en una devolución de indicación de 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-ACTION-CONFIRMED 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 ninguna otra 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. Además, 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 un tipo de BOOLEAN.

<i>Error específico</i>	<i>Código</i>
failure	00000001
<i>Parámetros</i>	<i>Referencias</i>
failureType	2.1.1.3.1
traceSent	2.1.1.3.1

<i>Parámetro</i>	<i>Código</i>
failureType	10000000
<i>Bit</i>	<i>Significado</i>
0	detectedLoop
1	excessiveLengthRoute
2	unknownObjectInstance
3	routeInaccessible
4	processingFailure
5	unknownInitiatingSP
6	timerExpired
7	sPNotAnSTP

<i>Parámetro</i>	<i>Código</i>
traceSent	10000001
<i>Contenido</i>	<i>Significado</i>
TRUE	se envió la información del rastreo
FALSE	no se envió la información del rastreo

failure	SPECIFIC-ERROR		
	PARAMETER SEQUENCE	{ failureType	[0] IMPLICIT FailureString,
		traceSent	[1] IMPLICIT BOOLEAN }
	::= 1		

FailureString ::= BIT STRING
{detectedLoop (0),
excessiveLengthRoute (1),
unknownObjectInstance (2),
routeInaccessible (3),
processingFailure (4),
unknownInitiatingSP (5),
timerExpired (6),
sPNotAnSTP (7)}

### 2.1.1.3.2 Partial Success (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>
partialSuccess	00000010
<i>Parámetros</i>	<i>Referencias</i>
failureType	2.1.1.3.1
traceSent	2.1.1.3.1

partialSuccess	SPECIFIC-ERROR		
	PARAMETER SEQUENCE	{ failureType	[0] IMPLICIT FailureString,
		traceSent	[1] IMPLICIT BOOLEAN }
	::= 2		



#### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulado como «PointCode», el contenido indica el código de punto exacto	2.1.1.1.4

#### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulado como «PointCode», el contenido indica el código de punto exacto	2.1.1.1.4

#### 2.1.2.1.4 unknownObjectInstance (instancia de objeto desconocido)

Si la instancia de objeto es desconocida, no se necesita información adicional.

<i>Parámetro</i>	<i>Código</i>
unknownObjectInstance	10000011
<i>Contenido</i>	<i>Referencia</i>
vacío	–

#### 2.1.2.1.5 routeInaccessible

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

<i>Parámetro</i>	<i>Código</i>
routeInaccessible	10000100
<i>Contenido</i>	<i>Referencia</i>
Bit 0 contiene el primer bit del código de punto Bit 1 contiene el segundo bit del código de punto, etc.	2.1.1.1.1

### 2.1.2.1.6 processingFailure

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

<i>Parámetro</i>	<i>Código</i>
processingFailure	10000101
<i>Contenido</i>	<i>Referencia</i>
vacío	–

### 2.1.2.1.7 unknownInitiatingSP

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

<i>Parámetro</i>	<i>Código</i>
unknownInitiatingSP	10000110
<i>Contenido</i>	<i>Referencia</i>
Bit 0 contiene el primer bit del código de punto Bit 1 contiene el segundo bit del código de punto, etc.	2.1.1.1.1

### 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.

<i>Parámetro</i>	<i>Código</i>
timerExpired	10100111
<i>Contenido</i>	<i>Referencia</i>
Secuencia de uno o más códigos de puntos rotulados como «PointCode», el contenido indica el código de punto exacto	2.1.1.1.4

### 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.

<i>Parámetro</i>	<i>Código</i>
sPNotAnSTP	10101000
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulado como «PointCode», el contenido indica el código de punto exacto	2.1.1.1.4

### 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 el macro CNF-ACTION definido en la Figura 3, y el suceso routeTrace se especifica mediante el macro 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 testRoute Action (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
<i>ActionArg</i>		<i>O/M</i>	<i>Referencias</i>
initiatingSP		M	3.1.1.1.1
traceRequested		M	3.1.1.1.2
threshold		M	3.1.1.1.3
pointCodesTraversed		M	3.1.1.1.4
formIndicator		M	3.1.1.1.5
mtpBackwardRoutingRequested		M	3.1.1.1.6
testInitiatorGT		O	3.1.1.1.7
destinationPC		O	3.1.1.1.8
destinationSSN		O	3.1.1.1.9
backupDPC		O	3.1.1.1.10
backupSSN		O	3.1.1.1.11
originalGT		O	3.1.1.1.12
<i>ActionResult</i>			
vacío			
<i>Linked Operations</i>			
N/A			
<i>Specific Errors</i>			<i>Referencias</i>
failure			3.1.1.3.1
partialSuccess			3.1.1.3.2

```

testRoute CNF-ACTION
ACTIONINFOARG SEQUENCE {
    initiatingSP                [0] IMPLICIT PointCode,
    traceRequested              [1] IMPLICIT BOOLEAN,
    threshold                   [2] IMPLICIT INTEGER,
    pointCodesTraversed        [3] IMPLICIT PointCodeList,
    formIndicator               [4] IMPLICIT FormIndicator,
    mtpBackwardRoutingRequested [5] IMPLICIT BOOLEAN,
    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
}
SPECIFICERRORS                {failure, partialSuccess}
 ::= 1

```

**3.1.1.1 testRoute Action Arguments (argumentos de 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.	

PointCode ::= OCTET STRING

**3.1.1.1.2 traceRequested (rastreo solicitado)**

traceRequested indica que se debe informar al originador sobre el rastreo de todas las rutas utilizadas para llegar al destino (routeTrace Event se describe en 3.1.2). 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) 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
<i>Contenido</i>	
Número entero	

### 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	

PointCodeList ::= SEQUENCE OF PointCode
---

### 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	

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

### 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 = Indicador de título global Octetos 2, 3, ... = Título global del iniciador	

GlobalTitle ::= OCTET STRING

### 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.	

SubsystemNumber ::= OCTET STRING

### 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 punto 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 SRVT 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 = Indicador de título global	
Octetos 2, 3,... = Título global original	

### 3.1.1.2 Action Results (resultados de acciones)

No hay contenido en una indicación de devolución de 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-ACTION-CONFIRMED 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>Parámetros</i>		<i>Referencias</i>
failureType	3.1.1.3.1	
traceSent	3.1.1.3.1	

<i>Parámetro</i>	<i>Código</i>
failureType	10000000
<i>Bit</i>	<i>Significado</i>
0	detectedLoop
1	excessiveLengthRoute
2	unknownObjectInstance
3	routeInaccessible
4	processingFailure
5	unknownInitiatingSP
6	timerExpired
7	wrongSP
8	incorrectTranslation-Primary
9	incorrectTranslation-Secondary
10	incorrectTranslation-Intermediate
11	notPrimaryDestination
12	notSecondaryDestination
13	notRecognizedPrimary
14	notRecognizedSecondary
15	routingProblem

<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

failure	SPECIFIC-ERROR PARAMETER SEQUENCE ::= 1	{ failureType traceSent	[0] IMPLICIT FailureString, [1] IMPLICIT BOOLEAN }
---------	---	----------------------------	---

```

FailureString ::= BIT STRING
                { detectedLoop (0),
                  excessiveLengthRoute (1),
                  unknownObject (2),
                  routeInaccessible (3),
                  processingFailure (4),
                  unknownInitiatingSP (5),
                  timerExpired (6),
                  wrongSP (7),
                  incorrectTranslation-Primary (8),
                  incorrectTranslation-Secondary (9),
                  incorrectTranslation-Intermediate (10),
                  notPrimaryDestination (11),
                  notSecondaryDestination (12),
                  notRecognizedPrimary (13),
                  notRecognizedSecondary (14),
                  routingProblem (15) }

```

**3.1.1.3.2 partialSuccess (éxito parcial)**

Esta indicación se presenta cuando por lo menos una invocación testRoute CnfAction 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
<i>Parámetros</i>	<i>Referencias</i>
failureType	3.1.1.3.1
traceSent	3.1.1.3.1

```

partialSuccess  SPECIFIC-ERROR
                PARAMETER SEQUENCE      { failureType  [0] IMPLICIT FailureString,
                                          traceSent     [1] IMPLICIT BOOLEAN }
                ::= 2

```

**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
<i>EventInfo</i>		O/M (Nota)	<i>Referencias</i>
success		O	3.1.2.1.1
detectedLoop		O	3.1.2.1.2
excessiveLengthRoute		O	3.1.2.1.3
unknownObjectInstance		O	3.1.2.1.4
routeInaccessible		O	3.1.2.1.5
processingFailure		O	3.1.2.1.6
unknownInitiatingSP		O	3.1.2.1.7
timerExpired		O	3.1.2.1.8
wrongSP		O	3.1.2.1.9
incorrectTranslation-Primary		O	3.1.2.1.10
incorrectTranslation-Secondary		O	3.1.2.1.11
incorrectTranslation-Intermediate		O	3.1.2.1.12
notPrimaryDestination		O	3.1.2.1.13
notSecondaryDestination		O	3.1.2.1.14
notRecognizedPrimary		O	3.1.2.1.15
notRecognizedSecondary		O	3.1.2.1.16
routingProblem		O	3.1.2.1.17
NOTA – Sólo debe estar presente una de estas indicaciones.			

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

### 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.

<i>Parámetro</i>	<i>Código</i>
success	10100000
<i>Contenido</i>	<i>Referencia</i>
Secuencia de uno o más códigos de puntos, rotulados «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

#### 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.

<i>Parámetro</i>	<i>Código</i>
detectedLoop	10100001
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

#### 3.1.2.1.3 excessiveLengthRoute (ruta demasiado larga)

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

<i>Parámetro</i>	<i>Código</i>
excessiveLengthRoute	10100010
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

#### 3.1.2.1.4 unknownObjectInstance (instancia de objeto desconocido)

Si el objeto 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.

<i>Parámetro</i>	<i>Código</i>
unknownObjectInstance	10000011
<i>Contenido</i>	<i>Referencia</i>
vacío	–

### 3.1.2.1.5 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
<i>Contenido</i>	<i>Referencia</i>
Bit 0 contiene el primer bit del código de punto, Bit 1 contiene el segundo bit del código de punto, etc.	3.1.1.1.1

### 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
<i>Contenido</i>	<i>Referencia</i>
vacío	–

### 3.1.2.1.7 unknownInitiatingSP (PS iniciador desconocido)

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

<i>Parámetro</i>	<i>Código</i>
unknownInitiatingSP	10000110
<i>Contenido</i>	<i>Referencia</i>
Bit 0 contiene el primer bit del código de punto Bit 1 contiene el segundo bit del código de punto, etc.	3.1.1.1.1

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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>
notRecognizedPrimary	10101110
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

### 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 de traducción.

<i>Parámetro</i>	<i>Código</i>
routingProblem	10110000
<i>Contenido</i>	<i>Referencia</i>
Secuencia de códigos de puntos, rotulados como «PointCode», el contenido indica el código de punto exacto	3.1.1.1.4

## 4 Gestión de circuitos

### 4.1 Elemento de servicio de aplicación de la prueba de validación de circuitos

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-action-confirmed 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 Cnf Action (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	<i>Temporizador = <math>T_c</math></i>	<i>Clase = 1</i>	<i>Código = 00000001</i>
<i>ActionArg</i>		<i>O/M</i>	<i>Referencias</i>
requestingSP		M	4.1.1
timer		O	4.1.1
<i>ActionResult</i>			<i>Referencia</i>
success			4.1.1
<i>Linked Operations</i>			
N/A			
<i>Specific Errors</i>			
failure			

cktValidTest	CNF-ACTION		
ACTIONARG SEQUENCE {			
requestingSP		RequestingSP,	
timer		Timer OPTIONAL }	
ACTIONRESULT			Success
SPECIFICERRORS			{ failure }
::= 3			

#### 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  
}
```

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.

## 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 ::= { ccitt 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). Véase el Anexo C/X.208 y 22/X.209.

Los Cuadros 1 y 2 muestran las primitivas OM; en la Figura 1 aparecen las operaciones OMAP obtenidas de CMIP (IS 9596) y la Figura 3 ilustra una sintaxis formal de OMASE.

A continuación figuran las operaciones obtenidas de CMIP.

Operaciones definidas actualmente	
0	eventReport
7	confirmedAction

## 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.

ManagedObjetClass: 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».

### **6.2.2.2.2 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».

### **6.2.2.2.3 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».

### **6.2.2.2.4 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).

**Servicio OM-CONFIRMED-ACTION**

Nombre de parámetro	Pet/Ind	Res/Con
CallingPartyAddress	M	M
CalledPartyAddress	M	M
DialogueID	M	M
InvokeID	M	M
AccessControl	O	–
BaseManagedObjectClass	M	–
BaseManagedObjectInstance	M	–
ActionInfo	M	–
ActionResult	–	M <sup>a)</sup>
ActionError		M <sup>b)</sup>
Timer	M <sup>c)</sup>	–

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 transactionId 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  $T_1$  para MRVT,  $T_2$  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, son ignorados por el OMPM.

#### **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 TC<sup>3)</sup>.

#### 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 TC<sup>3)</sup>.

#### 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

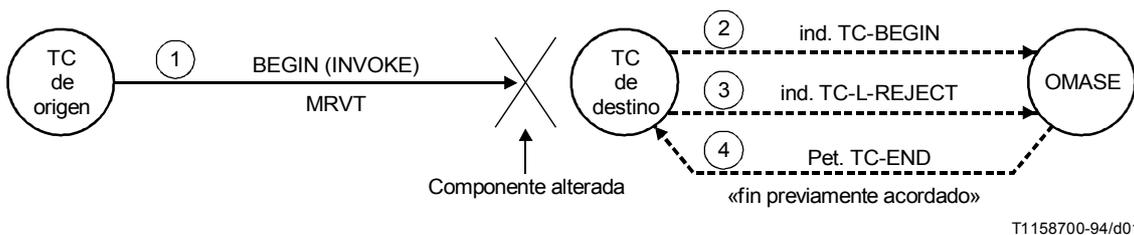


FIGURA 2a/Q.754

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 «ruta inaccesible».

Esto se ilustra en la Figura 2b.

#### 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 «ruta inaccesible».

#### 6.2.3.2.9 Recepción de la indicación TC-P-ABORT

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

<sup>3)</sup> 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.

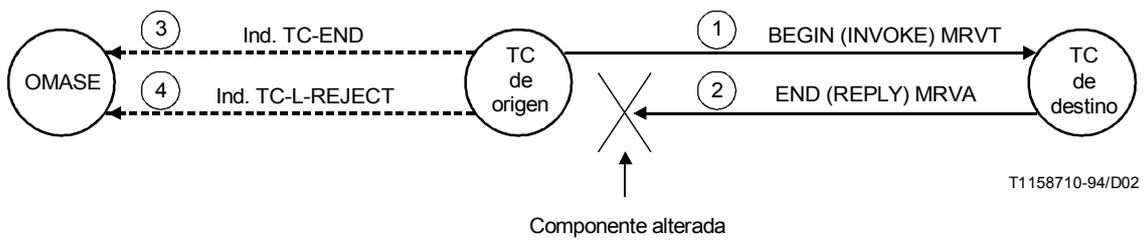


FIGURA 2b/Q.754

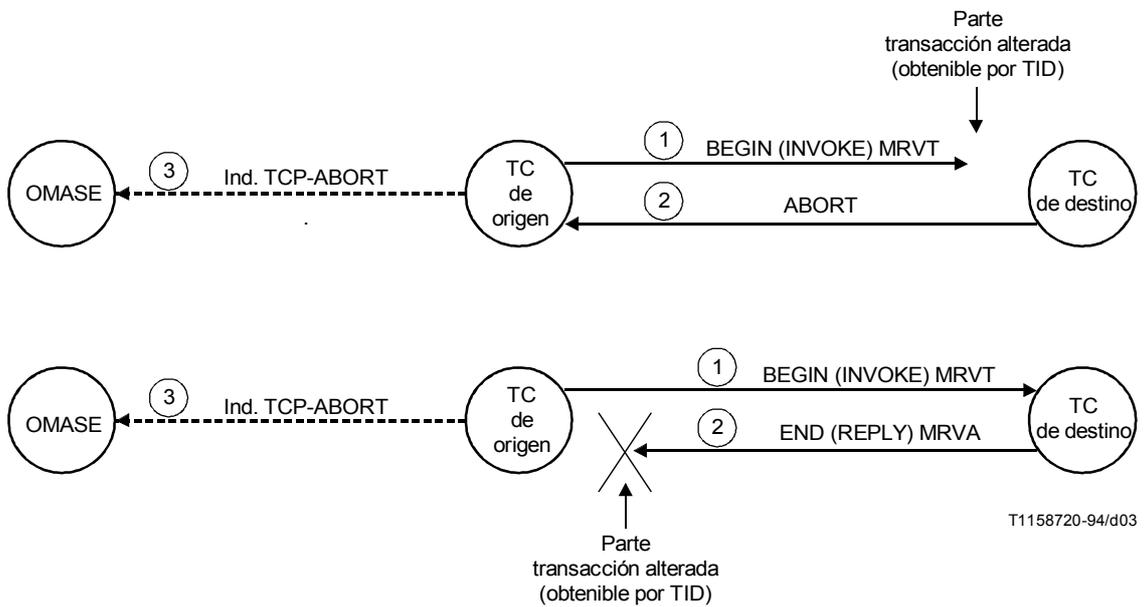


FIGURA 2c/Q.754

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 «ruta inaccesible».

### 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 «ruta inaccesible».

## 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

-- protocolo OMASE

**OMASE { ccitt recommendation q 754 omase(0) version1(1) }**

**DEFINITIONS ::=**

**BEGIN**

-- definiciones TCAP

**EXPORTS EVENT, CNF-ACTION, SPECIFIC-ERROR;**

**IMPORTS OPERATION, ERROR FROM TCAPMessage { ccitt recommendation q 773 modules(0) messages(1) version2(2) };**

-- operaciones OMASE

**eventReport OPERATION**

**PARAMETER SEQUENCE {**

<b>managedObjectClass</b>	<b>ObjectClass,</b>
<b>managedObjectInstance</b>	<b>ObjectInstance,</b>
<b>eventTime</b>	<b>[5] IMPLICIT GeneralizedTime OPTIONAL,</b>
<b>eventType</b>	<b>[7] IMPLICIT EVENT,</b>
<b>eventInfo</b>	<b>[8] ANY DEFINED BY eventType OPTIONAL }</b>

**::= localValue 0**

**confirmedAction OPERATION**

**PARAMETER SEQUENCE {**

<b>COMPONENTS OF</b>	<b>BaseManagedObjectId,</b>
<b>accessControl</b>	<b>[5] AccessControl OPTIONAL,</b>
<b>actionInfo</b>	<b>[12] IMPLICIT ActionInfo }</b>

**RESULT actionResult                    ActionResult**

**ERRORS { accessDenied, invalidArgumentValue,  
noSuchAction, noSuchArgument,  
noSuchObjectClass, noSuchObjectInstance,  
processingFailure }**

**::= localValue 7**

FIGURA 3/Q.754 (hoja 1 de 5)

Sintaxis formal de servicios OMASE

-- Las definiciones de error de servicio om son las siguientes:

<b>noSuchObjectClass</b> <b>PARAMETER</b> <b>::= localValue 0</b>	<b>ERROR</b> <b>ObjectClass</b>
<b>noSuchObjectInstance</b> <b>PARAMETER</b> <b>::= localValue 1</b>	<b>ERROR</b> <b>ObjectInstance</b>
<b>accessDenied</b> <b>::= localValue 2</b>	<b>ERROR</b>
<b>noSuchAction</b> <b>PARAMETER</b> <b>::= localValue 9</b>	<b>ERROR</b> <b>NoSuchAction</b>
<b>processingFailure</b> <b>PARAMETER</b> <b>::= localValue 10</b>	<b>ERROR</b> <b>ProcessingFailure -- opcional</b>
<b>noSuchArgument</b> <b>PARAMETER</b> <b>::= localValue 14</b>	<b>ERROR</b> <b>NoSuchArgument</b>
<b>invalidArgumentValue</b> <b>PARAMETER</b> <b>::= localValue 15</b>	<b>ERROR</b> <b>InvalidArgumentValue</b>

FIGURA 3/Q.754 (hoja 2 de 5)  
**Sintaxis formal de servicios OMASE**

-- A continuación se indican las definiciones de tipo de soporte:

<b>ActionArgument</b>	<b>::= SEQUENCE</b>	<b>{ COMPONENTS OF accessControl actionInfo</b>	<b>BaseManagedObjectId, [5] AccessControl OPTIONAL, [12] IMPLICIT ActionInfo }</b>
<b>ActionInfo</b>	<b>::= SEQUENCE</b>	<b>{ actionTypes actionInfoArg</b>	<b>ActionTypeId, [4] ANY DEFINED BY actionTypes OPTIONAL }</b>
<b>ActionResult</b>	<b>::= SEQUENCE</b>	<b>{ managedObjectClass  managedObjectInstance currentTime  actionReply</b>	<b>ObjectClass OPTIONAL, ObjectInstance OPTIONAL, [5] IMPLICIT GeneralizedTime OPTIONAL, [6] IMPLICIT ActionReply OPTIONAL }</b>
<b>ActionTypeId</b>	<b>::= CHOICE</b>	<b>{ -- globalForm... localForm</b>	<b>[3] IMPLICIT CNF-ACTION }</b>
<b>BaseManagedObjectId</b>	<b>::= SEQUENCE</b>	<b>{ baseManagedObjectClass baseManagedObjectInstance</b>	<b>ObjectClass, ObjectInstance }</b>
<b>EventReportArgument</b>	<b>::= SEQUENCE</b>	<b>{ managedObjectClass managedObjectInstance eventTime  eventType eventInfo</b>	<b>ObjectClass, ObjectInstance, [5] IMPLICIT GeneralizedTime OPTIONAL, EventTypeId, [8] ANY DEFINED BY eventType OPTIONAL }</b>
<b>EventTypeId</b>	<b>::= CHOICE</b>	<b>{ -- globalForm... localForm</b>	<b>[7] IMPLICIT EVENT }</b>

FIGURA 3/Q.754 (hoja 3 de 5)

**Sintaxis formal de servicios OMASE**

<b>InvalidArgumentValue</b>	<b>::= CHOICE</b>	<b>{ actionValue eventValue eventType eventInfo</b>	<b>[0] IMPLICIT ActionInfo, [1] IMPLICIT SEQUENCE { EventTypeId, [8] ANY DEFINED BY eventType OPTIONAL }</b>
<b>NoSuchAction</b>	<b>::= SEQUENCE</b>	<b>{ managedObjectClass actionType</b>	<b>ObjectClass, ActionTypeId}</b>
<b>NoSuchArgument</b>	<b>::= CHOICE</b>	<b>{ actionId managedObjectClass actionType eventId managedObjectClass eventType</b>	<b>[0] IMPLICIT SEQUENCE { ObjectClass OPTIONAL, ActionTypeId }, [1] IMPLICIT SEQUENCE{ ObjectClass OPTIONAL, EventTypeId }</b>
<b>ObjectClass</b>	<b>::= CHOICE</b>	<b>{ globalForm  --... }</b>	<b>[0] IMPLICIT OBJECT IDENTIFIER,</b>
<b>ObjectInstance</b>	<b>::= CHOICE</b>	<b>{ --... nonSpecificForm --... }</b>	<b>[3] IMPLICIT OCTET STRING,</b>
<b>ProcessingFailure</b>	<b>::= SEQUENCE</b>	<b>{ managedObjectClass  managedObjectInstance specificErrorInfo</b>	<b>ObjectClass OPTIONAL, ObjectInstance OPTIONAL, [5] IMPLICIT SpecificErrorInfo }</b>
<b>SpecificError</b>	<b>::= INTEGER</b>	<b>-- definido por clase de objeto</b>	
<b>SpecificErrorInfo</b>	<b>::= SEQUENCE</b>	<b>{ errorType errorParm</b>	<b>[0] IMPLICIT SpecificError, [1] ANY DEFINED BY errorType OPTIONAL }</b>
<b>Timer</b>	<b>::= INTEGER</b>	<b>-- segundos</b>	

FIGURA 3/Q.754 (hoja 4 de 5)  
**Sintaxis formal de servicios OMASE**

-- Los informes de sucesos específicos se categorizan por clase de objeto. Las utilizaciones del protocolo pueden describirse en el EVENT MACRO que figura a continuación.

**EVENT MACRO ::=**

**BEGIN**

<b>TYPE NOTATION</b>	<b>::=</b>	<b>EventInfo</b>
<b>VALUE NOTATION</b>	<b>::=</b>	<b>value(VALUE INTEGER)</b>
<b>EventInfo</b>	<b>::=</b>	<b>"EVENTINFO" NamedType   empty</b>
<b>NamedType</b>	<b>::=</b>	<b>identifier type   type</b>

**END**

-- Las acciones específicas se categorizan por clase de objeto. Las utilizaciones del protocolo pueden describirse por el macro CNF-ACTION que figura a continuación.

**CNF-ACTION MACRO ::=**

**BEGIN**

<b>TYPE NOTATION</b>	<b>::=</b>	<b>ActionArg ActionResult SpecificErrors</b>
<b>VALUE NOTATION</b>	<b>::=</b>	<b>value(VALUE INTEGER)</b>
<b>ActionArg</b>	<b>::=</b>	<b>"ACTIONARG" NamedType   empty</b>
<b>ActionResult</b>	<b>::=</b>	<b>"ACTIONRESULT" NamedType   empty</b>
<b>SpecificErrors</b>	<b>::=</b>	<b>"SPECIFICERRORS" "{" SpecificErrorList"}"   empty</b>
<b>NamedType</b>	<b>::=</b>	<b>identifier type   type</b>
<b>SpecificErrorList</b>	<b>::=</b>	<b>SpecificError   SpecificErrorList", "SpecificError</b>
<b>SpecificError</b>	<b>::=</b>	<b>value(SPECIFIC-ERROR)</b>

**END**

-- Los errores que son específicos de una acción o un suceso se definen mediante el macro SPECIFIC-ERROR que figura a continuación.

**SPECIFIC-ERROR MACRO ::=**

**BEGIN**

<b>TYPE NOTATION</b>	<b>::=</b>	<b>ProcessingErrorParm</b>
<b>VALUE NOTATION</b>	<b>::=</b>	<b>value(VALUE INTEGER)</b>
<b>ProcessingErrorParm</b>	<b>::=</b>	<b>"PARAMETER" NamedType   empty</b>
<b>NamedType</b>	<b>::=</b>	<b>identifier type   type</b>

**END**

**END -- protocolo OMASE**

FIGURA 3/Q.754 (hoja 5 de 5)

**Sintaxis formal de servicios**

## Anexo A

(Este anexo es parte integrante de esta Recomendación)

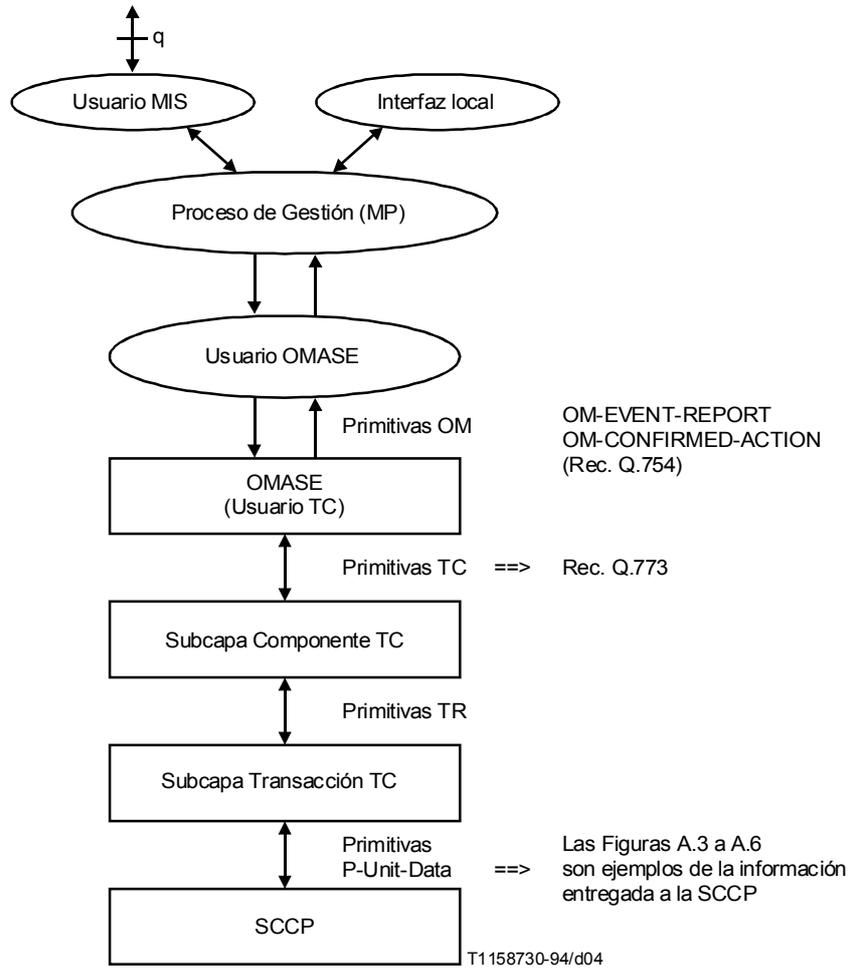


FIGURA A.1/Q.754  
**Interfaz de las primitivas**

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.

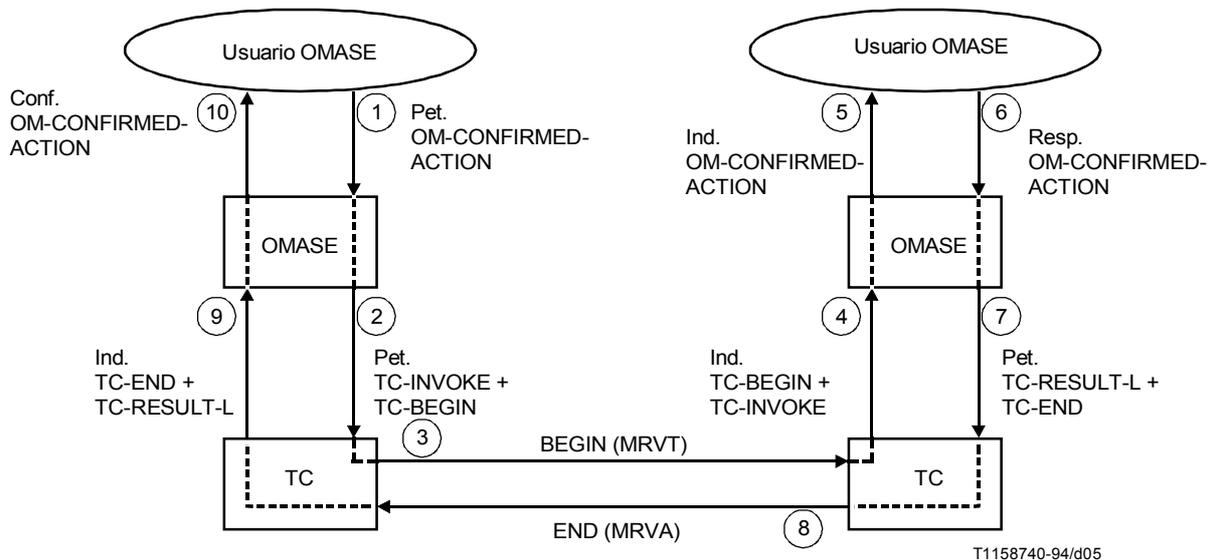


FIGURA A.2/Q.754  
Ejemplo de utilización de los interfaces de primitiva

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	xxxxxxx xxxxxxx xxxxxxx	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	xxxxxxx	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.209
Longitud	00000101	5 octetos
Valores – Tablas de encaminamiento por la MTP	00000000	CCITT, Rec.
	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.209
Longitud	00000010	2 octetos
Valor instancia de objeto	xxxxxxx xxxxxxx	Destino probado (SMAP)
Rótulo de información de acción	10101100	Recs X.711 y X.209
Longitud	00010011	Los 19 octetos que siguen aquí
Rótulo de tipo de acción	10000011	Forma local Recs. X.711 y X.209
Longitud	00000001	1 octeto
CNF-ACTION	00000001	= testRoute (8.1.1/Q.754)
Rótulo de Arg de info de acción	10100100	Recs. X.711 y X.209
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	8.1.1.1/Q.754, X.209
Longitud	00000010	2 octetos
Valor de SP iniciador	xxxxxxx xxxxxxx	(OMAP) iniciador de prueba
Rótulo de petición de rastreo	10000001	8.1.1.1.2/Q.754, X.209
Longitud	00000001	1 octeto
Valor	00000001	= TRUE
Rótulo de umbral	10000010	= umbral (8.1.1.1.3/Q.754)
Longitud	00000001	1 octeto
Valor de umbral	xxxxxxx	Proporcionado por el OMAP
Rótulo de código de punto atrav.	10100011	8.1.1.1.4/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	Igual que en COMENZAR (BEGIN) (mensaje MRVT)
	xxxxxxx	
	xxxxxxx	
	xxxxxxx	
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	Igual que en Comienzo (BEGIN) (mensaje MRVT)
	xxxxxxx	
	xxxxxxx	
	xxxxxxx	
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	2.1.1.3.1/Q.754
Parámetros de error	10100001	2.1.1.3.1/Q.754
Longitud	00000111	Los 7 octetos que siguen aquí
Rótulo de tipo de fallo	10000000	2.1.1.3.1/Q.754
Longitud	00000010	2 octetos
Bits no utilizados	00000000	Ningún bit
Cadena de fallo	xxxxxxx	Depende del tipo de fallo (2.1.1.3.1/Q.754)
Rótulo de rastreo enviado	10000001	2.1.1.3.1/Q.754
Longitud	00000001	1 octeto
Valor de rastreo enviado	0000000x	True = 1, False = 0 (2.1.1.3.1/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 00000000	CCITT, Rec. q 85 => 754 72 => Tablas de encaminamiento por la MTP 1992
Rótulo de instancia de objeto	10000011	(Figura 3/Q.754)
Longitud	00000010	2 octetos
Valor de instancia de objeto	xxxxxxx xxxxxxx	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 (2.1.2/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	2.1.2.1.1/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**





