



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.606.1

(02/2003)

SERIE X: REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS

Gestión de redes de interconexión de sistemas abiertos y
aspectos de sistemas – Gestión de redes

**Tecnología de la información – Protocolo
perfeccionado de transporte de
comunicaciones: Especificación de la
gestión de la calidad de servicio en el
transporte multidifusión símplex**

Recomendación UIT-T X.606.1

RECOMENDACIONES UIT-T DE LA SERIE X
REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999

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

**Tecnología de la información – protocolo perfeccionado de transporte
de comunicaciones: Especificación de la gestión de la calidad
de servicio en el transporte multidifusión simplex**

Resumen

En esta Recomendación | Norma Internacional (la segunda parte del protocolo de transporte de comunicaciones perfeccionado (ECTP, *enhanced communications transport protocol*)) se especifica un protocolo de transporte multidifusión extremo a extremo para apoyar las funciones de negociación, supervisión y mantenimiento de la calidad de servicio (QoS, *quality of service*) de una conexión multidifusión (1-a-n) simplex. Las aplicaciones multidifusión, como los servicios multimedia de flujo continuo en tiempo real, pueden utilizarlo para soportar sus requisitos de QoS.

Si la gestión QoS está habilitada, la negociación de los parámetros QoS se puede hacer opcionalmente durante la fase de creación de la conexión. Entonces, el emisor controlador arbitra los valores de los parámetros a partir de los que proponen los receptores. Si la negociación no está habilitada, el emisor impone valores predefinidos.

El nivel QoS alcanzado se supervisa durante la fase de transferencia de datos. Cada receptor mide los valores QoS propios y comunica el estado de cada parámetro a su controlador padre mediante paquetes de acuse de recibo (ACK, *acknowledgment*) modificados. El emisor agrega los valores de estado de los parámetros comunicados por los receptores para obtener una visión del estado QoS de la conexión. El emisor, basándose en el análisis del estado QoS de la conexión, puede tomar medidas de mantenimiento QoS, como ajustar la velocidad de transmisión de los datos.

Orígenes

La Recomendación UIT-T X.606.1, preparada por la Comisión de Estudio 17 (2001-2004) del UIT-T, fue aprobada el 13 de febrero de 2003. Se publica también un texto idéntico como Norma Internacional ISO/CEI 14476-2.

Palabras clave

Protocolo perfeccionado de transporte de comunicaciones (ECTP), protocolo de transporte multidifusión, multidifusión simplex, calidad de servicio (QoS).

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 Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), 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 1 de la AMNT.

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 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 2003

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

ÍNDICE

Página

1	Alcance.....	1
2	Referencias normativas	2
3	Definiciones	2
3.1	Términos definidos en la Rec. UIT-T X.605 ISO/CEI 13252	2
3.2	Términos definidos en la Rec. UIT-T X.606 ISO/CEI 14476-1	2
3.3	Términos definidos en esta Recomendación Norma Internacional	2
4	Acrónimos	2
4.1	Tipos de paquetes	3
4.2	Varios	3
5	Convenciones	3
6	Visión general	3
7	Componentes de la gestión QoS.....	6
7.1	Elemento información de conexión.....	6
7.2	Parámetros QoS.....	7
7.3	Elemento extensión QoS	8
7.4	Elemento acuse de recibo.....	9
7.5	Paquetes utilizados para la gestión QoS.....	10
8	Procedimientos para la gestión QoS.....	10
8.1	Negociación QoS	11
8.1.1	Procedimientos de negociación.....	11
8.1.2	Negociación QoS en la jerarquía del árbol.....	12
8.1.3	Negociación MSS	12
8.1.4	Reserva de recursos.....	13
8.2	Supervisión QoS.....	13
8.2.1	Generación de paquete ACK.....	13
8.2.2	Medición de los valores de parámetro QoS.....	14
8.2.3	Correspondencia con el valor de estado de parámetro	14
8.2.4	Envío de informes al emisor	15
8.3	Mantenimiento QoS	16
8.3.1	Ajuste de la velocidad de transmisión de datos.....	16
8.3.2	Pausa y reinicio de la conexión	17
8.3.3	Expulsión del problemático.....	18
8.3.4	Terminación de la conexión	18
9	Temporizadores y variables.....	18
9.1	Temporizadores.....	18
9.2	Variables de funcionamiento.....	18
Anexo A	– Interfuncionamiento de ECTP y RSVP para la reserva de recursos.....	20
A.1	Parámetros QoS de ECTP	20
A.2	Visión general de RSVP.....	20
A.2.1	RSVP SENDER_TSPEC	20
A.2.2	ADSPEC de RSVP.....	21
A.2.3	RSVP FLOWSPEC.....	21
A.2.4	RSVP API	21
A.3	Ejemplo de correspondencia de parámetros RSVP con parámetros ECTP	22
A.4	Caso de interfuncionamiento de ECTP y RSVP	22
Anexo B	– Interfaces de programación de aplicación	25
B.1	Visión general	25
B.1.1	Funciones API.....	25
B.1.2	Utilización de las funciones ECTP API	25

	<i>Página</i>
B.2 Funciones ECTP API	26
B.2.1 msocket().....	26
B.2.2 mbind().....	27
B.2.3 maccept().....	28
B.2.4 mconnect()	29
B.2.5 msend()	30
B.2.6 mrecv().....	30
B.2.7 mclose().....	31
B.2.8 mgetsockopt() y msetsockopt()	31
B.3 Ejemplo del fichero de encabezamiento msocket.h.....	33
Bibliografía.....	37

Introducción

Esta Recomendación | Norma Internacional especifica el protocolo perfeccionado de transporte de comunicaciones (ECTP, *enhanced communications transport protocol*), que es un protocolo de transporte diseñado para soportar aplicaciones multidifusión en Internet, que funcionan en redes con capacidad de multidifusión. El ECTP funciona en redes IPv4/IPv6 capaces de transmitir con el protocolo Internet para multidifusión con la ayuda de los protocolos de enrutamiento del protocolo Internet para gestión de grupos (IGMP, *Internet group management protocol*) y el protocolo Internet para multidifusión (*IP multicast*), como se muestra en la figura 1. Cabe la posibilidad de aplicar el ECTP en el protocolo de datagramas de usuario (UDP, *user datagram protocol*).

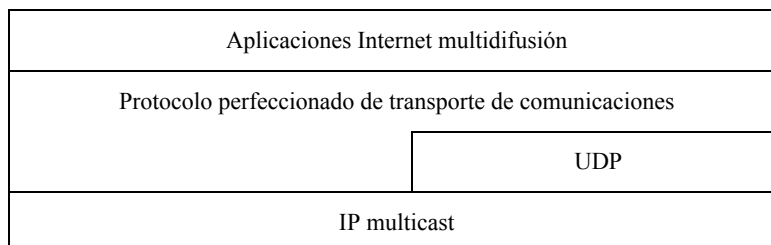


Figura 1 – Modelo ECTP

El objetivo del ECTP es soportar conexiones multidifusión estrictamente controladas en aplicaciones simplex, dúplex y N-plex. Esta parte del ECTP (parte 2) especifica las funciones de gestión QoS para lograr una gestión equilibrada de la QoS de las conexiones multidifusión simplex de los usuarios. La funcionalidad de gestión QoS consiste en las operaciones negociación QoS, supervisión QoS y mantenimiento QoS. Los procedimientos del protocolo para el control de la fiabilidad en el transporte multidifusión simplex se definen en la parte 1 de ECTP (Rec. UIT-T X.606 | ISO/CEI 14476-1), que es parte integrante de esta Recomendación | Norma Internacional. En otras partes de la norma se definirán los procedimientos de control y las funciones de gestión QoS correspondientes al caso dúplex (X.ectp-3 | ISO/CEI 14476-3 y X.ectp-4 | ISO/CEI 14476-4) y el caso N-plex (X.ectp-5 | ISO/CEI 14476-5 y X.ectp-6 | ISO/CEI 14476-6).

En el ECTP, todos los posibles miembros se enroлан en un grupo multidifusión antes de que se cree la conexión o la sesión. Esos miembros definen el grupo enrolado. Cada receptor de un grupo enrolado se denomina "receptor enrolado". En el proceso de enrolamiento se autentica cada miembro. La información sobre el grupo, como la clave de grupo, las direcciones IP multidifusión y los números de puerto, se distribuyen a los miembros enrolados durante el proceso de enrolamiento. Para los miembros del grupo enrolado se crea una conexión ECTP.

El emisor es el elemento central de las comunicaciones de grupo multidifusión. A un único emisor de la conexión multidifusión simplex se le asigna el papel de propietario de la misma, que en esta especificación se denomina "propietario superior" (TO, *top owner*). El propietario de la conexión es responsable de la gestión global de la misma, y dirige las operaciones de creación, terminación pausa y reinicio de la conexión, así como las operaciones de incorporación tardía al y separación del grupo.

El emisor inicia el proceso de creación de la conexión. Algunos o todos los receptores enrolados participan en la conexión, y se los designa "receptores activos". Los receptores activos en esta fase pueden participar en la negociación de la calidad de servicio deseada para la sesión. Cualquier receptor enrolado que no esté activo en esta fase puede participar en la conexión como miembro tardío, pero tendrá que aceptar la QoS establecida. Un receptor activo puede abandonar la conexión.

Una vez creada la conexión, el emisor empieza a transmitir datos multidifusión. Mientras la conexión está activa, el emisor monitorea el estado de la sesión mediante los paquetes de control que envían los receptores activos con información sobre el proceso a efectos de retroalimentación (información de retroalimentación).

El emisor puede tomar diversas medidas si la información recibida de los receptores activos indica problemas en la red (como congestión grave). Esas medidas consisten en ajustar la velocidad de transmisión de datos, suspender la transmisión de datos multidifusión temporalmente o, como último recurso, terminar la conexión.

Esta especificación de la gestión QoS se utiliza en las aplicaciones multidifusión que quieren soportar diversos requisitos de QoS, y en los modelos de facturación/tarificación correspondientes.

**NORMA INTERNACIONAL
RECOMENDACIÓN UIT-T**

**Tecnología de la información – protocolo perfeccionado de transporte
de comunicaciones: Especificación de la gestión de la calidad
de servicio en el transporte multidifusión símplex**

1 Alcance

Esta Recomendación | Norma Internacional es parte integrante de la Rec. UIT-T X.606.x | ISO/CEI 14476 "ECTP: Protocolo perfeccionado de transporte de comunicaciones", que es una familia de especificaciones de los protocolos diseñados para soportar servicios de transporte multidifusión.

La Rec. UIT-T X.606 | ISO/CEI 14476-1 ofrece la especificación de diversas operaciones de protocolo para el transporte multidifusión símplex. Las operaciones de protocolo son la gestión de la conexión, como creación/terminación de la conexión y su pausa/reinicio; la gestión de los miembros, como los miembros que se incorporan tardíamente o se abandonan el grupo y el seguimiento de esas actividades; y el control de los errores de transporte de datos multidifusión, como la detección y la recuperación de errores.

Esta parte de la Recomendación | Norma Internacional ofrece la especificación de la gestión QoS para lograr una calidad de servicio deseable en la conexión de transporte multidifusión símplex.

Esta especificación describe las siguientes operaciones de gestión QoS:

a) *Negociación QoS*

Con respecto a la negociación QoS, en esta especificación se supone que el nivel QoS deseado para un servicio de aplicación multidifusión se puede expresar mediante un conjunto de parámetros QoS. La negociación QoS se hace por medio de un intercambio de paquetes de control entre el emisor y los receptores. El emisor propone los valores objetivo de los parámetros QoS indicados por los requisitos de la aplicación, y luego cada receptor puede proponer valores modificados basándose en su sistema y/o su capacidad de red. El emisor arbitra los valores modificados propuestos por los receptores. Los valores objetivo de los parámetros QoS se pueden utilizar como parámetros de entrada para reservar los recursos de red.

b) *Supervisión QoS*

En ECTP, el control QoS se basa en la información de los paquetes de control que envían los receptores. Los mensajes de los receptores con información de realimentación le permiten al emisor seguir el número de receptores activos y, también, monitorear el estado de la conexión para el transporte de datos multidifusión. La supervisión QoS tiene por objetivo que el emisor pueda diagnosticar el estado de la conexión en términos de los valores de los parámetros QoS para, entonces, tomar las medidas necesarias a fin de mantener el estado de la conexión en el nivel QoS deseado. El estado de la conexión supervisada se comunica a la aplicación en el lado del emisor. La información transmitida puede proporcionar estadísticas útiles para, por ejemplo, la facturación.

c) *Mantenimiento QoS*

Basándose en la información de realimentación de los receptores, el emisor toma una o más medidas para mantener el estado de la conexión al nivel QoS deseado. Estas medidas de mantenimiento QoS pueden ser el ajuste de la velocidad de transmisión de los datos, la pausa y reinicio de la conexión, la expulsión del miembro problemático (el problemático) y la terminación de la conexión. Estas funciones de supervisión QoS y mantenimiento QoS, que se basan en el estado de los parámetros supervisados, permiten controlar la congestión que ocasiona la velocidad de la transmisión.

Esta Recomendación | Norma Internacional es parte integrante de la Rec. X.606.x | ISO/CEI 14476, que consta de 6 partes. Todos los componentes del protocolo, incluidos los formatos de paquete y los procedimientos de protocolo especificados en la Rec. UIT-T X.606 | ISO/CEI 14476-1, también son válidos en esta Recomendación | Norma Internacional.

2 Referencias normativas

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y las Normas citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

- Recomendación UIT-T X.601 (2000), *Marco para comunicaciones entre múltiples pares*.
- Recomendación UIT-T X.605 (1998) | ISO/CEI 13252:1999, *Tecnología de la información – Definición del servicio perfeccionado de transporte de comunicaciones*.
- Recomendación UIT-T X.606 (2001) | ISO/CEI 14476-1: 2002, *Tecnología de la información – Protocolo perfeccionado de transporte de comunicaciones: Especificación del transporte multidifusión simplex*

3 Definiciones

3.1 Términos definidos en la Rec. UIT-T X.605 | ISO/CEI 13252

Esta Recomendación | Norma Internacional se basa en los conceptos desarrollados en la norma servicio perfeccionado de transporte de comunicaciones (ECTS) (Rec. UIT-T X.605 | ISO/CEI 13252)

- a) parámetros QoS;
- b) negociación QoS; y
- c) arbitraje QoS.

3.2 Términos definidos en la Rec. UIT-T X.606 | ISO/CEI 14476-1

Esta Recomendación | Norma Internacional se describe sobre la base de los conceptos y términos desarrollados en la especificación del transporte multidifusión simplex con ECTP, Rec. UIT-T X.606 | ISO/CEI 14476-1).

- a) aplicación;
- b) paquete;
- c) emisor;
- d) receptor;
- e) árbol;
- f) padre; e
- g) hijo.

3.3 Términos definidos en esta Recomendación | Norma Internacional

En esta Recomendación | Norma Internacional se aplican las definiciones siguientes:

- a) **supervisión QoS**: operación del protocolo que se utiliza para diagnosticar el estado actual de la conexión. En la supervisión o monitoreo QoS cada receptor tiene que medir los valores de parámetro propios y comunicarlos al emisor. El emisor agrega la información de estado comunicada por los receptores.
- b) **mantenimiento QoS**: operación del protocolo que se utiliza para mantener el estado de la conexión en el nivel QoS deseado. El emisor toma medidas de mantenimiento QoS basándose en la información de estado supervisada.

4 Acrónimos

A los efectos de esta Recomendación | Norma Internacional se aplican las siglas siguientes.

4.1 Tipos de paquetes

ACK	Acuse de recibo (<i>acknowledgment</i>)
CC	Confirmación creación de conexión (<i>connection creation confirm</i>)
CR	Petición creación de conexión (<i>connection creation request</i>)
CT	Terminación de conexión (<i>connection termination</i>)
DT	Datos (<i>data</i>)
HB	Latido (<i>heartbeat</i>)
JC	Confirmación incorporación tardía (<i>late join confirm</i>)
JR	Petición incorporación tardía (<i>late join request</i>)
LR	Petición separación (<i>leave request</i>)
ND	Datos nulos (<i>null data</i>)
RD	Datos de retransmisión (<i>retransmission data</i>)

4.2 Varios

API	Interfaces de programación de aplicaciones (<i>application programming interfaces</i>)
CHQ	Calidad más alta controlada (<i>controlled highest quality</i>)
Diffserv	Servicios diferenciados (<i>differentiated services</i>)
ECTP	Protocolo perfeccionado de transporte de comunicaciones (<i>enhanced communications transport protocol</i>)
ECTS	Servicio perfeccionado de transporte de comunicaciones (<i>enhanced communications transport service</i>)
IP	Protocolo Internet (<i>Internet protocol</i>)
LQA	Calidad más baja admisible (<i>lowest quality allowed</i>)
MSS	Tamaño máximo de segmento (<i>maximum segment size</i>)
OT	Objetivo de funcionamiento (<i>operating target</i>)
QoS	Calidad de servicio (<i>quality of service</i>)
RSVP	Protocolo de reserva de recursos (<i>resource reservation protocol</i>)

5 Convenciones

En esta Recomendación | Norma Internacional, los grados de obligación del cumplimiento de los requisitos para que las implementaciones se conformen a la norma ECTP se expresan así: debe(n) (must), requerido (required), deberá(n) (shall), no deben (must not), no deberán (shall not), deberían (should), no deberían (should not), puede(n) (may), y opcional (optional).

6 Visión general

Esta Recomendación | Norma Internacional ofrece la especificación de la gestión QoS para las conexiones de transporte multidifusión en modo uno a muchos (simplex). Se describen las funciones de gestión QoS siguientes:

- 1) negociación QoS, incluida la reserva de recursos de red;
- 2) supervisión QoS; y
- 3) mantenimiento QoS.

En la fase de creación de la conexión, el emisor informa a los receptores si la gestión QoS está habilitada. Cuando está habilitada, el emisor especifica también si la negociación QoS se realizará o no en la conexión. Las operaciones de supervisión QoS y mantenimiento QoS se hacen únicamente cuando la gestión QoS está habilitada.

En la figura 2 se ilustran estas funciones de la gestión QoS para la conexión multidifusión simplex. Las funciones del protocolo indicadas con línea punteada se especifican en la Rec. UIT-T X.600 | ISO/CEI 14476-1.

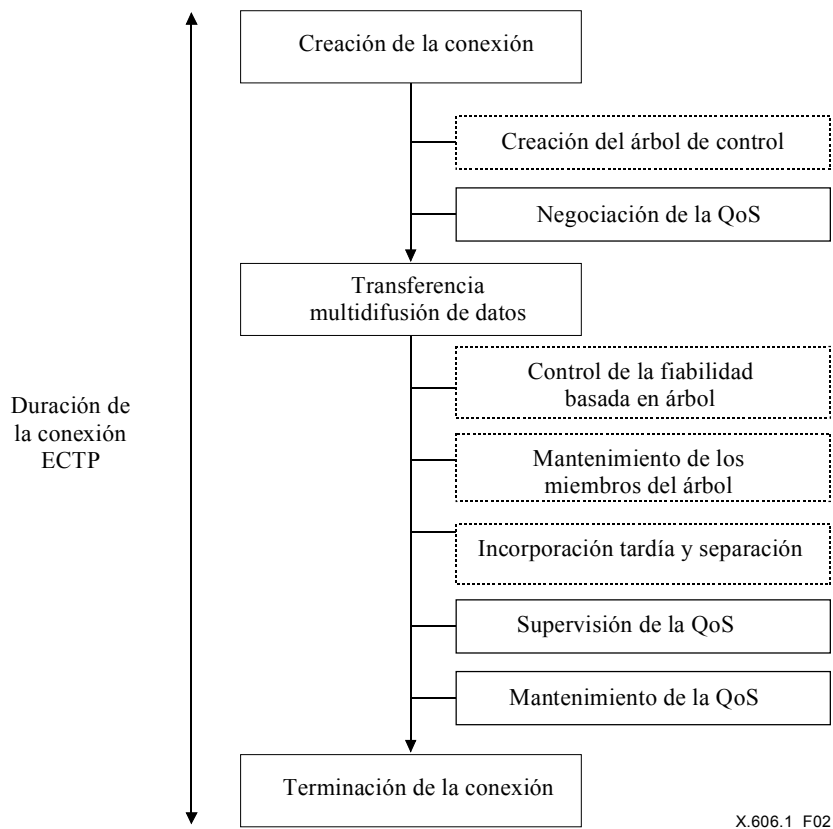


Figura 2 – Gestión QoS en ECTP

En general, la QoS es la calidad de servicio obligatoria para recibir satisfactoriamente los datos de una aplicación en el receptor para, por ejemplo, lograr la calidad de visualización de audio/vídeo deseable. En esta especificación, se supone que los requisitos de QoS de una aplicación se expresan en términos de uno o más parámetros QoS; caudal, retardo de tránsito, fluctuación de fase del retardo de tránsito y tasa de pérdida de datos. Según los requisitos de la aplicación, puede que en la conexión no se utilicen todos estos parámetros QoS. Por ejemplo, un servicio en tiempo no real no tiene por qué imponer el requisito de retardo de tránsito.

A partir de los requisitos de las aplicaciones, el emisor determina los valores objetivo de cada parámetro QoS. La correspondencia de los requisitos de la aplicación con los valores objetivo de los parámetros queda fuera del alcance de esta Recomendación, pero se puede realizar mediante programas de aplicación.

La negociación QoS se realiza en la fase de creación de la conexión. El emisor propone los valores objetivo deseados para cada parámetro QoS a todos los receptores mediante multidifusión. Para caudal se especifican tres valores objetivo: calidad más alta controlada (CHQ, *controlled highest quality*), objetivo de funcionamiento (OT, *operating target*) y calidad más baja admisible (LQA, *lowest quality allowed*). Para el resto de los parámetros, retardo de tránsito, fluctuación de fase del retardo de tránsito y tasa de pérdida de datos, sólo se especifican dos valores objetivo: OT y LQA.

Cuando la negociación QoS está habilitada, cada receptor puede proponer modificaciones a los valores de parámetro propuestos por el emisor. Para hacer una modificación se tendrá en cuenta la capacidad del sistema en el lado receptor y los entornos de red. Se imponen las restricciones siguientes a las modificaciones que realicen los receptores en los parámetros:

- 1) los receptores no modificarán los valores OT;
- 2) los valores modificados por los receptores estarán comprendidos dentro de los valores LQA y CHQ propuestos por el emisor.

Los valores de parámetro modificados por los receptores se transmiten al emisor mediante mensajes ACK. El emisor arbitra los distintos valores enviados por los receptores y elige de común acuerdo una gama de valores.

La figura 3 es un esquema abstracto de la negociación QoS que se puede producir en ECTP. A partir de los requisitos de la aplicación, el emisor configura un conjunto de valores objetivo de parámetro QoS. El emisor comunica a los receptores los valores objetivo (paso 1). Basándose en esos valores objetivo, cada receptor empieza a reservar recursos

con ayuda del protocolo de reserva de recursos (RSVP, *resource reservation protocol*) o de servicios diferenciados (Diffserv, *differentiated services*) (paso 2). Si la negociación QoS está habilitada en la conexión, cada receptor puede proponer valores modificados de los parámetros QoS (paso 3). A partir de esos valores modificados, el emisor arbitra los valores y determina valores arbitrados (paso 4). Los valores arbitrados se entregan al receptor por medio de paquetes de latido (HB) o de confirmación incorporación tardía (JC), y se utiliza luego en la supervisión y el mantenimiento QoS.

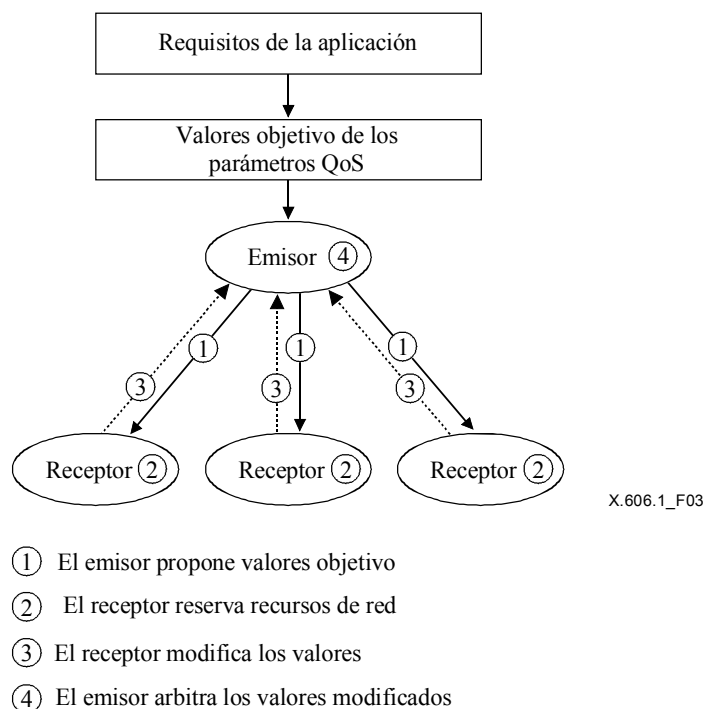


Figura 3 – Negociación QoS

Una vez creada la conexión ECTP, y si la gestión QoS está habilitada, se realizan las operaciones supervisión QoS y mantenimiento QoS para la transmisión de los datos multidifusión. En cuanto a la supervisión QoS, cada receptor tiene que medir los propios valores de parámetro. Basándose en los valores medidos y en los valores negociados, el receptor determina el valor de estado del parámetro para cada parámetro como un entero: normal (0), razonable (1), posiblemente anormal (2) o anormal (3). Estos valores de estado se entregan al emisor mediante paquetes ACK.

El emisor agrega los valores de estado de los parámetros comunicados por los receptores. Si se emplea un árbol de control, cada nodo propietario local (LO, *local owner*) padre agrega los valores medidos comunicados por sus hijos, y retransmite el o los valores agregados a su propio padre utilizando paquetes ACK.

En la figura 4 se ilustran las operaciones supervisión QoS y mantenimiento QoS descritas en esta especificación.

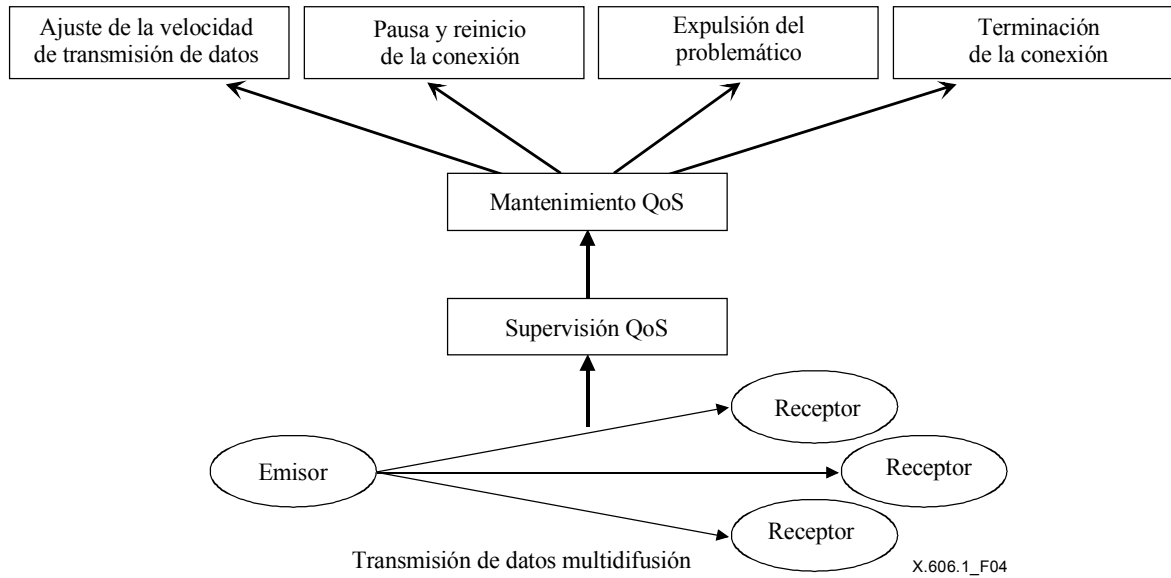


Figura 4 – Esquema conceptual de la supervisión QoS y el mantenimiento QoS

El emisor toma las medidas de mantenimiento QoS necesarias para mantener el estado de la conexión en el nivel de QoS deseado, basándose en los valores de estado supervisados. Se dispone de reglas específicas preconfiguradas para iniciar las medidas de mantenimiento QoS, como el ajuste de velocidad de transmisión de datos, la pausa y reinicio de la conexión, la expulsión del miembro que causa problemas (problemático) y la terminación de la conexión. Se observa cuántos receptores están en estado anormal o posiblemente anormal y se toman las medidas.

7 Componentes de la gestión QoS

En esta cláusula se describe los componentes del protocolo ECTP obligatorios para las operaciones de la gestión QoS. Todos los componentes definidos en la Rec. UIT-T X.606 | ISO/CEI 14476-1 se han extendido.

7.1 Elemento información de conexión

En la figura 5 se muestra el elemento información de conexión especificado en la Rec. UIT-T X.606 | ISO/CEI 14476-1.

0		8	16	24	31
Elemento siguiente	Versión	Banderas	Opción de configuración de árbol	Nivel máximo de hojas	Número máximo de hojas
Tiempo de creación de la conexión			Tamaño de la correspondencia de bits de ACK		Reservada

Figura 5 – Elemento información de conexión

En la gestión QoS, el emisor ECTP especifica los tres campos siguientes en el byte "banderas":

7	6	5	4	3	2	1	0
Reservado	Retx	N	QoS	CT			

- 1) *QoS* – bit de bandera que indica si la gestión QoS está habilitada (1) o no habilitada (0) en la conexión. Cuando el bit se fija a "1" se invocan todos los procedimientos de la gestión QoS. El valor por defecto es "0";

- 2) *N (negociación)* – bit de bandera que indica si la negociación QoS está habilitada (1) o no habilitada (0) en la conexión. Cuando el bit se fija a "1", cada receptor puede proponer sus propios valores de parámetro. El valor por defecto es "0";
- 3) *Retx (retransmisión)* – bit de bandera que indica si el padre ejecuta retransmisiones (0) o no las ejecuta (1). Cuando el bit se fija a "1", el emisor o los padres no tienen obligación de retransmitir paquetes RD, incluso si se reciben peticiones retransmisión en los paquetes ACK. El valor por defecto es "0".

El bit QoS se tiene que fijar a "1" (QoS habilitada) antes de que el bit N sea válido. Hay tres casos posibles.

- a) Bit QoS fijado a "1" y bit N fijado a "0" indica que se va a utilizar QoS en la conexión y que el emisor impondrá los valores de QoS. Los receptores no pueden negociar los.
- b) Ambos bits fijados a "1" indica que se va a utilizar QoS en la conexión y que el emisor y los receptores pueden negociar los valores de parámetro QoS.
- c) Bit QoS fijado a "0" indica que no se va a utilizar QoS en la conexión. El bit N no se utiliza en este caso.

El bit de retransmisión Retx se puede fijar independientemente de la fijación del bit QoS. Se espera que las aplicaciones de medios de flujo continuo en directo y en tiempo real no necesiten recuperación de errores basada en la retransmisión, pero sí necesitan las funciones de gestión QoS. Aun en este caso, se utilizan los paquetes ACK para comunicar la información de estado de la conexión.

7.2 Parámetros QoS

En esta especificación se definen los cuatro parámetros QoS siguientes:

- 1) caudal (bytes por segundo);
- 2) retardo de tránsito (milisegundos);
- 3) fluctuación de fase del retardo de tránsito (milisegundos);
- 4) tasa de pérdida de datos (porcentaje).

El caudal representa una cantidad de salida de datos de aplicación promediada en un intervalo de tiempo. "Caudal objetivo" es el valor de caudal necesario para la visualización deseable de los datos de aplicación. Las aplicaciones generan datos multidifusión, y el emisor ECTP los transmite basándose en el o los valores de caudal objetivo. La velocidad real de recepción de los datos en el lado del receptor depende de la velocidad de transmisión de los datos, las condiciones de la red, la capacidad del sistema de extremo, etc.

Para caudal, el emisor configura los valores objetivo siguientes:

- 1) caudal CHQ;
- 2) caudal OT;
- 3) caudal LQA.

Son obligatorias las siguientes desigualdades: $\text{caudal LQA} \leq \text{caudal OT} \leq \text{caudal CHQ}$.

El retardo de tránsito representa el tiempo de transmisión extremo a extremo de un emisor a un receptor. Para lograr la visualización deseable de datos multidifusión, el emisor puede configurar los valores objetivo siguientes:

- 1) retardo de tránsito OT;
- 2) retardo de tránsito LQA.

Son obligatorias las siguientes desigualdades: $\text{retardo de tránsito OT} \leq \text{retardo de tránsito LQA}$.

La fluctuación de fase del retardo de tránsito representa las variaciones de los valores de retardo de tránsito. Para lograr la visualización deseable de los datos, el emisor puede configurar los valores objetivo siguientes:

- 1) fluctuación de fase del retardo de tránsito OT;
- 2) fluctuación de fase del retardo de tránsito LQA.

Son obligatorias las siguientes desigualdades: $\text{fluctuación de fase del retardo de tránsito OT} \leq \text{fluctuación de fase del retardo de tránsito LQA}$.

La tasa de pérdida de datos se define como la relación de la cantidad de datos perdidos y la cantidad total de datos transmitidos. Para lograr la visualización deseable de los datos, el emisor puede configurar los valores objetivo siguientes:

- 1) tasa de pérdida OT;
- 2) tasa de pérdida LQA.

Son obligatorias las siguientes desigualdades: tasa de pérdida OT $\leq$ tasa de pérdida LQA.

7.3 Elemento extensión QoS

Para la gestión QoS, en esta especificación se define por primera vez el elemento "extensión QoS". A continuación, se listan todos los elementos extensión utilizados en ECTP.

Cuadro 1 – Cuadro de codificación de los elementos extensión en ECTP

Elemento	Codificación
Información de conexión	0001
Acuse de recibo	0010
Árbol de miembros	0011
Indicación de tiempo	0100
QoS	0101
Ningún elemento	0000

El elemento extensión QoS especifica el tamaño máximo de segmento (*MSS*) y los valores objetivo de los parámetros QoS en ECTP descritos en 7.2. Como se muestra en la figura 6, el elemento QoS tiene una longitud de "28" bytes.

0	8	16	24	31
Elemento siguiente	Versión	Banderas QoS	Tamaño máximo de segmento	
Caudal CHQ				
Caudal OT				
Caudal LQA				
Retardo de tránsito OT			Retardo de tránsito LQA	
Fluctuación de fase del retardo de tránsito OT			Fluctuación de fase del retardo de tránsito LQA	
Tasa de pérdida OT		Tasa de pérdida LQA		Reservado

Figura 6 – Elemento extensión QoS

Se especifican los siguientes parámetros:

- Elemento siguiente* – Indica el tipo de elemento siguiente que sigue inmediatamente a este elemento QoS;
- Versión* – Define la versión actual de este elemento, empezando por "1";
- Banderas QoS* – Byte de bandera que especifica si se utiliza cada uno de los parámetros QoS y el tamaño máximo de segmento (*MSS*) en la conexión. En la siguiente figura se ilustra la codificación de este byte. Cuando un bit se fija a "1" se utiliza el parámetro QoS o el *MSS* correspondientes. El valor por defecto es "0" para cada uno de los bits.

7	6	5	4	3	2	1	0
Reservado	E	D	C	B	A		

- A – caudal;
- B – retardo de tránsito;
- C – fluctuación de fase del retardo de tránsito;
- D – tasa de pérdida de datos;
- E – tamaño máximo de segmento;
- Reservado* – se reserva para uso futuro.

- d) *Tamaño máximo de segmento (MSS)* – Representa el tamaño máximo de un segmento o paquete ECTP en unidades de byte. Cuando el bit "E" de las banderas QoS se fija a "1" hay que negociar el MSS (véase 8.1.3). De lo contrario, se utiliza 1024 como valor por defecto de MSS.
- e) *Valores de caudal* – Cada valor es un número entero sin signo de 32 bits expresado en bytes por segundo. Los siguientes valores objetivo son válidos sólo si el bit "A" de las banderas QoS se fija a "1";
 - 1) Caudal CHQ – Límite superior de caudal;
 - 2) Caudal OT – Caudal objetivo para la visualización deseada de los datos multidifusión;
 - 3) Caudal LQA – Límite inferior de caudal.
- f) *Valores de retardo de tránsito* – Cada valor es un número entero sin signo de 16 bits expresado en milisegundos. Los siguientes valores objetivo son válidos sólo si el bit "B" de las banderas QoS se fija a "1";
 - 1) Retardo de tránsito OT – Retardo de tránsito objetivo para la visualización deseada de los datos multidifusión;
 - 2) Retardo de tránsito LQA – Retardo de tránsito máximo admisible.
- g) *Valores de fluctuación de fase del retardo de tránsito* – Cada valor es un número entero sin signo de 16 bits expresado en milisegundos. Los siguientes valores objetivo son válidos sólo si el bit "C" de las banderas QoS se fija a "1":
 - 1) Fluctuación de fase del retardo de tránsito OT – Fluctuación de fase del retardo de tránsito objetivo para la visualización deseada de los datos multidifusión;
 - 2) Fluctuación de fase del retardo de tránsito LQA – Fluctuación de fase del retardo de tránsito máxima admisible.
- h) *Valores de tasa de pérdida de datos* – Cada valor es un número entero sin signo de 8 bits, en una gama de 0 a 100, expresado en porcentaje. Los siguientes valores objetivo son válidos sólo si el bit "D" de las banderas QoS se fija a "1".
 - 1) Tasa de pérdida OT – Tasa de pérdida objetivo para la visualización deseada de los datos multidifusión;
 - 2) Tasa de pérdida LQA – Tasa de pérdida máxima admisible.
- i) *Reservado* – Se reserva para uso futuro.

Se utiliza el elemento QoS para que el emisor comunique a los receptores los valores objetivo de los parámetros QoS mediante un paquete de petición creación de conexión (CR) en la fase de creación de la conexión. Durante la negociación QoS, el elemento QoS se utiliza también cuando el receptor propone sus propios valores modificados al emisor. Los valores de QoS negociados se dan a conocer a los miembros tardíos mediante el paquete de confirmación incorporación tardía (JC) y a los miembros activos, mediante el paquete latido (HB).

También se hace referencia a estos valores de QoS en las operaciones supervisión QoS y mantenimiento QoS.

7.4 Elemento acuse de recibo

En la supervisión QoS, cada receptor tiene que medir los propios valores de parámetro. El valor de parámetro medido se hace corresponder con el valor de estado de parámetro. El valor de estado es un número entero, como 0, 1, 2 ó 3. Cuanto más grande es el valor de estado, peor es el estado de la conexión.

Los valores de estado se entregan al emisor mediante paquetes ACK. El elemento acuse de recibo del paquete ACK contiene los valores de estado de los parámetros QoS que se utilizan en la conexión.

A continuación, se muestra el elemento acuse de recibo especificado en la Rec. UIT-T Rec. X.606 | ISO/CEI 14476-1. En la figura, el byte "estado de parámetro" se define por primera vez en esta Especificación.

0	8	16	24	31
Elemento siguiente	Versión	Longitud de la correspondencia de bits válida	Reservado	Estado de parámetro
Número más bajo de secuencia				
Correspondencia de bits				

Figura 7 – Elemento acuse de recibo

El byte "estado de parámetro" tiene la siguiente estructura:

7	6	5	4	3	2	1	0
D	C	B	A				

- a) *A* – representa dos bits que indican el valor de estado del caudal medido;
- b) *B* – representa dos bits que indican el valor de estado del retardo de tránsito medido;
- c) *C* – representa dos bits que indican el valor de estado de la fluctuación de fase del retardo de tránsito medida;
- d) *D* – representa dos bits que indican el valor de estado de la tasa de pérdida de paquetes medida

Un valor de estado que consta de dos bits tiene uno de los siguientes valores:

- a) 00 – indica "0" como valor de estado;
- b) 01 – indica "1" como valor de estado;
- c) 10 – indica "2" como valor de estado;
- d) 11 – indica "3" como valor de estado.

En 8.2.3 se describen con detalle los esquemas de correspondencia de valor de parámetro medido con valor de estado.

7.5 Paquetes utilizados para la gestión QoS

En el cuadro 2 se enumeran los paquetes ECTP que se utilizan en la gestión QoS.

Cuadro 2 – Paquetes ECTP utilizados en la gestión QoS

Tipo de paquete	Elemento extensión				
	Información de conexión	Árbol de miembros	Acuse de recibo	Indicación de tiempo	QoS
CR	O				O
CC		O			O
HB		O		O	O
ACK		O	O	O	
JC	O				O

El paquete CR contiene un elemento QoS. El emisor utiliza este elemento para proponer (o imponer) los valores objetivo de los parámetros QoS que se utilizan en la conexión. Se puede hacer referencia a estos valores con los mecanismos y protocolos de reserva de recursos, como RSVP, si están habilitados en la red. Cuando la negociación QoS está habilitada, cada receptor responde al emisor con sus propios valores propuestos para los parámetros QoS por intermedio de un paquete de confirmación creación de conexión (CC). El emisor arbitra las propuestas devueltas, y envía a los receptores los valores arbitrados de los parámetros QoS mediante paquetes HB. En el caso de un miembro tardío, los valores objetivo de parámetro QoS que se están utilizando en la conexión (impuestos o negociados originalmente) se notifican mediante el paquete JC (véase 8.1).

En las operaciones de supervisión QoS y mantenimiento QoS se hace referencia a los valores objetivo o los valores negociados. Se utilizan paquetes ACK para comunicar los valores de estado de los parámetros QoS observados en el lado receptor (véase 8.2).

8 Procedimientos para la gestión QoS

En ECTP, la gestión QoS incluye las siguientes funciones:

- 1) negociación QoS, posiblemente con reserva de recursos de red;
- 2) supervisión QoS;
- 3) mantenimiento QoS.

La negociación QoS se realiza durante la fase de creación de la conexión, mientras que las funciones supervisión QoS y mantenimiento QoS se efectúan en la fase de transmisión de datos.

Si la gestión QoS está habilitada en la conexión, las funciones supervisión QoS y mantenimiento QoS se ejecutan por defecto. Por otro lado, la negociación QoS se habilita sólo cuando el bit *N* del bit "banderas" del elemento información de conexión se fija a "1".

8.1 Negociación QoS

El emisor ECTP transmite un paquete CR a todos los receptores para iniciar la fase de creación de conexión. El paquete CR contiene los valores objetivo (propuestos o impuestos) de cada parámetro QoS, como CHQ, OT y LQA. Cada uno de los receptores puede hacer referencia a estos valores objetivo para la reserva de recursos (véase 8.1.4). Si la negociación QoS está habilitada en la conexión, se activan los procedimientos de negociación (véase 8.1.1). Los valores objetivo impuestos o negociados se utilizan luego en la supervisión QoS y el mantenimiento QoS (véanse 8.2 y 8.3).

Cuando la negociación QoS está habilitada, cada receptor puede proponer un nuevo valor modificado en respuesta a un valor objetivo de parámetro propuesto por el emisor. Para proponer un nuevo valor, el receptor tiene que poder identificar los recursos de sistema o de red que se van a utilizar. Por ejemplo, se puede evaluar un valor de caudal modificado a partir de las velocidades de línea de los enlaces de transmisión accesibles en el sitio del receptor (como línea de abonado digital (DSL, *digital subscriber line*), módem de cable, redes inalámbricas, etc.). El valor modificado también se puede determinar mediante el análisis de los requisitos de usuario de extremo en un sitio de receptor. Un *host* de extremo puede utilizar un programa informático para determinar el valor de parámetro modificado a efectos de negociación, a partir de los recursos de red y de sistema y de los requisitos de usuario de extremo. Sin embargo, en situaciones reales no es fácil identificar con exactitud la capacidad de los recursos de red que intervienen en un receptor. Por consiguiente, y al menos en un futuro próximo, la negociación QoS se efectuará basándose en los requisitos de usuario de extremo en el nivel de aplicación o en la capacidad de sistema del *host* de extremo.

En esta Especificación, el emisor tiene que especificar, por intermedio del elemento extensión QoS, si hay que negociar cada uno de los parámetros QoS (véase 7.3). Si los parámetros son negociables, el receptor puede proponer valores modificados. Cuando el receptor no desea modificar un parámetro QoS, simplemente devuelve el mismo elemento QoS que recibió del emisor.

8.1.1 Procedimientos de negociación

Cuando la negociación QoS está activada en la conexión, cada receptor responde al emisor con un paquete de confirmación creación de conexión (CC), que contiene los valores objetivo modificados de los parámetros QoS respectivos.

En esta subcláusula se describen los procedimientos de negociación QoS del parámetro "caudal", que tiene tres valores objetivo: LQA, OT y CHQ. Los procedimientos de negociación de los parámetros restantes, retardo, fluctuación de fase del retardo y tasa de pérdida, son iguales, salvo que estos parámetros no tienen valores CHQ.

Durante la negociación QoS, los receptores no modificarán el valor OT de cada parámetro.

Los procedimientos detallados de la negociación QoS se describen a continuación y se ilustran en la figura 8.

- 1) El emisor propone los valores objetivo de parámetro:

A partir de los requisitos de la aplicación, el emisor determina los valores objetivo de parámetro: LQA_o, OT_o, CHQ_o, donde LQA_o < OT_o < CHQ_o, y a continuación envía un paquete CR con el elemento extensión QoS a todos los receptores.

- 2) Los receptores modifican los valores de parámetro:

En respuesta a los valores objetivo propuestos por el emisor, cada receptor *i* puede proponer valores modificados: LQA_i y CHQ_i. El valor OT_o no se modificará. Así, son obligatorias las siguientes desigualdades para cada receptor *i*: LQA_o < LQA_i < OT_o < CHQ_i < CHQ_o. Cada uno de los receptores envía los valores modificados al emisor mediante un paquete CC.

- 3) El emisor arbitra los valores de parámetro modificados:

El emisor arbitra los valores de parámetro modificados propuestos por los receptores como sigue:

$$CHQ_{min} = \min CHQ_i, \text{ para cada receptor } i;$$

$$LQA_{max} = \max LQA_i, \text{ para cada receptor } i.$$

LQA_{max} y CHQ_{min} son los valores de parámetro negociados resultantes de la negociación QoS.

- 4) El emisor anuncia los valores de parámetro negociados:

El emisor anuncia los valores LQA_{max}, CHQ_{min} y OT a los receptores mediante paquetes HB y JC.

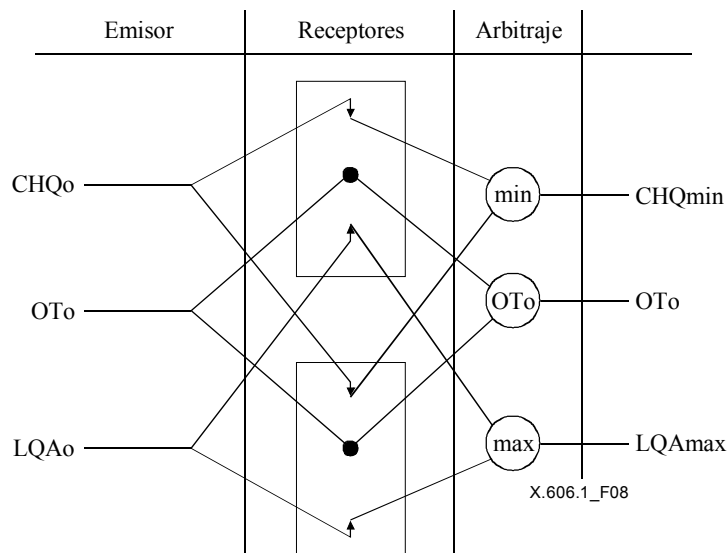


Figura 8 – Procedimientos de la negociación QoS

Para los parámetros retardo, fluctuación de fase del retardo y tasa de pérdida se obtiene un valor LQA en lugar de LQAmay, ya que en estos parámetros no se utilizan valores CHQ y valor $OT < \text{valor LQA}$, o sea

$$LQA_{\min} = \min LQA_i, \text{ para cada receptor } i.$$

No se ejecuta la negociación QoS para los receptores de incorporación tardía. El emisor se limita a notificar los valores negociados a los receptores tardíos mediante un paquete JC.

8.1.2 Negociación QoS en la jerarquía del árbol

Si se emplea un árbol de control con más de dos niveles, cada LO padre tiene que efectuar los procedimientos de arbitraje QoS de los valores modificados propuestos por sus hijos (paso 3 de 8.1.1). En este caso, la regla de arbitraje es;

$$CHQ_{\min}' = \min CHQ_i, \text{ para cada hijo } i,$$

$$LQA_{\max}' = \max LQA_i, \text{ para cada hijo } i.$$

El padre envía a su padre los valores arbitrados, $CHQ_{\min}'$ y $LQA_{\max}'$, en un paquete CC. Entonces, el emisor recibe paquetes CC de todos sus hijos, y a continuación obtiene los valores arbitrados correspondientes a todos los receptores.

8.1.3 Negociación MSS

El MSS representa el tamaño máximo de paquete, que depende de la unidad máxima de transmisión (*MTU*, *maximum transmission unit*) en las transmisiones de capa de enlace. Los valores *MTU* típicos son 1500 bytes para Ethernet, 1492 bytes para IEEE 802, 4352 bytes para interfaz de datos distribuidos en fibra (FDDI, *fibre distributed data interface*), 576 bytes para X.25, etc. La *MTU* determina el tamaño de trama de las transmisiones de capa de enlace y, por consiguiente, el valor de MSS.

El valor *MSS* por defecto en ECTP es 1024 bytes, por lo que soporta la mayoría de los tipos de enlace, excepto X.25. Cuando el emisor y/o el receptor no pueden identificar su valor *MSS*, toman el valor *MSS* por defecto de 1024 bytes.

Cuando en ECTP se indica la negociación *MSS* (véase 7.3), se ejecutan los siguientes procedimientos:

- 1) El emisor escribe su valor *MSS* en el paquete CR, y lo transmite a todos los receptores.
- 2) Cada receptor propone su propio valor de *MSS* mediante el paquete CC. La *MTU* de la red local a la que pertenece el receptor afecta el *MSS*. Cuando el *MSS* del receptor es más largo que el del emisor, el receptor utiliza el *MSS* del emisor.
- 3) El emisor arbitra el *MSS* de la conexión y elige el valor *MSS* mínimo para todos los receptores.

8.1.4 Reserva de recursos

El ECTP por sí mismo no garantiza los niveles de QoS que necesitan las aplicaciones, pero se pueden utilizar los valores objetivo de los parámetros QoS para la reserva de los recursos de red, que se hace mediante los modelos de servicios integrados como protocolo de reserva de recursos (RSVP, *resource reservation protocol*) o servicios diferenciados (*Diffserv*).

El modelo RSVP se adapta bien al protocolo ECTP, ya que la reserva de recursos se hace de extremo a extremo. En las redes en las que RSVP está habilitado se entregan periódicamente mensajes RSVP PATH a los receptores. En respuesta al mensaje PATH, cada receptor envía periódicamente mensajes RSVP RESV al emisor por el trayecto de datos multidifusión.

Cuando se utiliza RSVP con ECTP, se hace referencia a los valores objetivo de los parámetros QoS de ECTP en la configuración de los descriptores de tráfico RSVP para RSVP TSPEC (o FLOWSPEC) contenido en el mensaje RSVP PATH. Estos mensajes PATH se envían de conformidad con el paquete CR y los paquetes HB transmitidos por el emisor. Se señala que los módulos de RSVP se gestionan independientemente del ECTP en los sistemas de extremo. Esto significa que la reserva de recursos y las funciones de control en RSVP sólo las ejecutan los *daemon* de RSVP correspondientes.

En el anexo A se describe un ejemplo de interfuncionamiento de ECTP y RSVP, junto con un posible esquema de correspondiente el de los parámetros de tráfico con los parámetros QoS asociados.

El modelo Servicio diferenciado ofrece clases de servicio diferenciadas para el tráfico IP, con objeto de soportar diversos tipos de aplicaciones y requisitos de empresa específicos. Se utiliza un pequeño patrón de bits en cada paquete, en el octeto de tipo de servicio (TOS, *type of service*) o punto de código de servicio diferenciado (DSCP, *differentiated service called point*) de IPv4, o en el octeto de clase de tráfico de IPv6, para marcar un paquete de modo que reciba un tratamiento de transmisión hacia delante particular, o un comportamiento por salto, en cada nodo de la red. Hay que estar familiarizado con la interpretación y utilización de este patrón de bits para el uso entre dominios y la interoperabilidad con múltiples proveedores, y tener una idea clara del comportamiento de los servicios previstos en una red. Hasta ahora, no se ha identificado un esquema de interfuncionamiento específico de ECTP y Diffserv.

El ECTP necesita utilizar mecanismos QoS de red subyacente, como RSVP y Diffserv, para establecer las conexiones de red que transportan los niveles de QoS obligatorios. No obstante, el uso potencial de ECTP no se limita a los métodos actuales de QoS de red. El ECTP se ha diseñado para que trabaje con QoS perfeccionada con arreglo a diversos modelos de QoS.

8.2 Supervisión QoS

La supervisión QoS proporciona al emisor información relativa al nivel de funcionamiento de la conexión. Para conocerlo, cada receptor tiene que medir los parámetros propios y comunicarlos al emisor.

En cuanto a los parámetros QoS utilizados en la conexión, cada receptor mide los valores propios. El valor medido se hace corresponder con el valor de estado de cada parámetro. El valor de estado de parámetro es un número entero 0, 1, 2 ó 3. Este valor de estado se consigna en el elemento acuse de recibo y se envía al emisor en el paquete ACK subsiguiente (véase 7.4). El emisor agrega los valores de estado de parámetro de todos los receptores.

La finalidad de la supervisión QoS es proporcionar al emisor información sobre el estado QoS de la conexión. Basándose en la información de estado supervisada, el emisor también puede tomar las medidas de mantenimiento QoS necesarias.

8.2.1 Generación de paquete ACK

Cada receptor comunica a su padre los *valores de estado de parámetro* mediante paquetes ACK. En ECTP-2, la regla de generación de paquete ACK es algo diferente de la especificada en ECTP-1. En ECTP-2, *AGT* se fija igual a *AGN*, es decir "*AGN* × 1 segundo" o "*AGN* segundos". Por consiguiente, cada receptor genera periódicamente paquetes ACK durante cada *AGT* segundos. La fijación de *AGN* es un asunto de implementación, pero en ECTP-2 se ha fijado a "8" por defecto.

Para generar el paquete ACK, cada receptor tiene que tener un temporizador que funciona en segundos, llamado "tiempo de supervisión QoS (QMT, *QoS monitoring time*). El temporizador QMT arranca no bien el receptor concluye el establecimiento de la conexión, o sea después de recibir el paquete CC o JC del emisor. El temporizador QMT aumenta monotónicamente conforme se realiza la conexión, es decir que no retorna a cero durante la conexión.

Cada receptor transmite un paquete ACK a su padre si:

$$QMT \% AGN = Child\ ID \% AGN$$

Se emplea este esquema para reducir en la mayor medida posible las implosiones de ACK en el lado padre. Mediante este mecanismo, cada receptor continuará generando paquetes ACK cada *AGN* segundos después del arranque del temporizador QMT. Por ejemplo, si *AGN* se fija a 8, un receptor con *Child ID* de 3 ó 11 generará paquetes ACK en los tiempos *QMT* de 3, 11, 19 y 27 segundos, etc. Se señala que en el ejemplo el primer ACK se genera para los paquetes de datos transmitidos sólo durante 3 segundos y no durante 8 segundos. Pero los demás paquetes ACK subsiguientes se generan para el intervalo de tiempo *AGN* segundos. De esta manera, cada receptor genera paquetes ACK cada *AGN* segundos, excepto para el primer paquete ACK.

Cada paquete ACK lleva el valor de estado de parámetro medido, como se describe en 8.2.2.

8.2.2 Medición de los valores de parámetro QoS

En la transmisión de datos multidifusión, cada receptor mide los valores propios de cada parámetro QoS. Todos los valores se miden, registran y calculan hasta que se genera un nuevo paquete ACK de conformidad con la regla de generación ACK descrita más arriba. Cuando llega el momento de enviar un paquete ACK, un receptor calcula el valor de estado de parámetro de los paquetes de datos recibidos y recopilados hasta entonces. Después de transmitir un ACK se suprimen los datos recopilados, y a continuación se recogen y consignan los nuevos datos para generar un nuevo ACK.

8.2.2.1 Caudal

El caudal se mide como la velocidad de recepción de datos expresada en unidades de bytes por segundo. La velocidad de recepción de datos se calcula mediante:

Cantidad de paquetes de datos recibidos, expresada en bytes, promediada en AGN segundos.

Una vez más, el primer paquete ACK se puede generar antes de que hayan transcurrido los *AGN* segundos. Para medir el valor de caudal, el receptor necesita conservar la información sobre cuántos paquetes de datos (bytes) se han recibido durante el intervalo específico.

Cada vez que se obtiene un valor de caudal nuevo se lo hace corresponder con el *valor de estado de parámetro*, un valor entero 0, 1, 2 ó 3, de conformidad con la regla de correspondencia descrita en 8.2.3.

8.2.2.2 Tasa de pérdida de datos

La tasa de pérdida de datos representa la tasa de pérdida de paquetes y se expresa en porcentaje. La tasa de pérdida de paquetes se calcula mediante:

Número de paquetes perdidos promediados en número de paquetes de datos recibidos durante AGN segundos.

Una vez más, el primer paquete ACK se puede generar antes de que hayan transcurrido *AGN* segundos. Para medir el valor de caudal, el receptor necesita conservar la información relativa a cuántos paquetes de datos se han perdido. Cuando el receptor genera un paquete ACK, el valor de la tasa de pérdida medido actualmente se hace corresponder con el *valor de estado de parámetro*.

8.2.2.3 Retardo de tránsito y fluctuación de fase del retardo de tránsito

Para medir un retardo de tránsito extremo a extremo, el emisor tiene que transmitir paquetes de datos con un elemento indicación de tiempo. Otro requisito es la sincronización de los relojes entre el emisor y los receptores. Si estos requisitos no se cumplen resulta difícil obtener información exacta sobre el retardo y la fluctuación de fase del tránsito. En esta subcláusula se supone que cada receptor puede medir el retardo de tránsito de extremo a extremo de un paquete de datos enviado por el emisor.

El retardo de tránsito se mide para cada paquete de datos recibido. Estos valores de retardo de tránsito se promedian en paquetes de datos recibidos durante *AGN* segundos. La fluctuación de fase del retardo de tránsito se mide como la diferencia entre los valores máximo y mínimo del retardo de tránsito de los paquetes de datos recibidos.

Durante los *AGN* segundos, cada vez que llega un nuevo paquete de datos se calcula el retardo de tránsito y se actualizan los valores promediados de retardo y de fluctuación de fase. Justo antes de que un receptor genere un paquete ACK, se hace corresponder el valor medido actualmente con el *valor de estado de parámetro*.

8.2.3 Correspondencia con el valor de estado de parámetro

El valor de parámetro medido se hace corresponder con el valor de estado de parámetro de cada parámetro QoS. El paquete ACK siguiente contiene, entonces, el valor de estado. Se señala una vez más que la medición de un parámetro QoS se activa únicamente cuando se indica la utilización del parámetro en la conexión (véase 7.3).

En la figura 9 se ilustra la correspondencia del valor medido con el valor de estado.

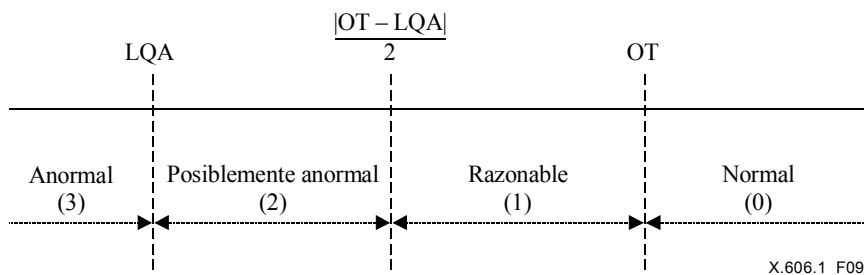


Figura 9 – Correspondencia del valor medido con el valor de estado

Como se muestra en la figura, la correspondencia del valor medido con el valor de estado se efectúa sobre la base de los valores de parámetro OT y LQA. El ECTP utiliza también un valor *umbral* para clasificar el estado como normal (0), razonable (1), posiblemente normal (2), y anormal (3). El valor *umbral* se fija a un valor mediano de los valores de parámetro OT y LQA (es decir, $|LQA - OT| / 2$), como se ilustra en la figura.

Un *valor de estado de parámetro* inicial se fija a "0". Una vez obtenido el valor medido, se efectúa la correspondencia con el *valor de estado de parámetro* como sigue:

SI "valor de parámetro medido > OT," entonces estado = 0,

PERO SI "umbral < valor de parámetro medido ≤ OT," entonces estado = 1,

PERO SI "LQA < valor de parámetro medido ≤ umbral " entonces estado = 2,

PERO SI "valor medido ≤ LQA, " entonces estado = 3.

En estas reglas de correspondencia sólo se mantienen las desigualdades para el parámetro caudal. Para el resto de los parámetros, retardo de tránsito, fluctuación de fase del retardo y tasa de pérdida, hay que invertir las desigualdades, porque valores OT ≤ valores LQA.

8.2.4 Envío de informes al emisor

Cada receptor comunica a su padre los *valores de estado de parámetro* obtenidos mediante paquetes ACK. Los paquetes ACK se generan sobre la base de *Child ID* y *AGN* segundos. De esta manera, toda la información relativa a los *valores de estado de parámetro* se envía al emisor siguiendo la jerarquía del árbol.

8.2.4.1 Agregación por LO en la jerarquía del árbol

En la jerarquía del árbol, cada LO padre agrega los paquetes ACK de sus hijos. Esta agregación se hace hasta justo antes de que el padre genere su propio paquete ACK. Además, el LO genera su propio paquete ACK cada *AGN* segundos, al igual que el receptor. Justo antes de generar paquetes ACK, el padre simplemente toma un valor promedio de los valores de estado de parámetro comunicados por sus hijos junto con su propio valor medido.

El valor promedio se calcula para cada parámetro QoS mediante

Suma de los valores de estado de parámetro de los hijos que responden promediados en CCN.

En la estructura de árbol multinivel, el número de descendientes se representa mediante el *número de receptores activos* (*ARN*, *active receiver number*) consignado en el elemento miembros del árbol. En este caso, los valores de estado de parámetro comunicados por los hijos se ponderan mediante sus valores *ARN* como sigue:

Valor promedio ponderado = $\sum_{i=1, \dots, C_{CN}} \{ARN(i) \times \text{valor de estado de parámetro}(i)\}$ sobre $\sum_{i=1, \dots, C_{CN}} ARN(i)$.

Un LO padre redondea este valor promedio a 0, 1, 2 ó 3 para cada parámetro QoS, y compone su paquete ACK.

8.2.4.2 Agregación por el emisor

La agregación de los valores de estado de parámetro por el emisor es igual a la agregación por LO. El emisor también lleva a cabo la agregación de los paquetes ACK comunicados por sus hijos cada *AGN* segundos.

Tras la agregación de los paquetes ACK, el emisor simplemente toma un valor promedio ponderado con el *ARN* de cada hijo. O sea:

Valor de estado agregado = suma de los valores de estado comunicados ponderada con el ARN de la conexión

El emisor obtiene un valor de estado agregado para cada parámetro QoS. Más específicamente, si se utiliza cada uno de los parámetros de la conexión, obtiene los siguientes valores de estado agregados:

- a) valor de estado agregado para caudal, indicado por *Tvalue*,
- b) valor de estado agregado para retardo de tránsito, indicado por *Dvalue*,
- c) valor de estado agregado para fluctuación de fase del retardo de tránsito, indicado por *Jvalue*,
- d) valor de estado agregado para tasa de pérdida de datos, indicado por *Lvalue*.

Cada uno de los valores agregados también está comprendido entre 0 y 3.

El emisor puede comunicar a la aplicación la información de estado supervisada. La información supervisada le sirve a la aplicación emisora para diagnosticar si la conexión funciona bien en términos de QoS, lo que a su vez sirve para diseñar el modelo de facturación/tarificación.

La información supervisada se utiliza también en el mantenimiento QoS. De entre los valores supervisados, *Lvalue* se usa para ajustar la velocidad de transmisión de los datos (véase 8.3.1). Se puede utilizar un valor de suma ponderada de todos los valores de estado, *Tvalue*, *Dvalue*, *Jvalue* y *Lvalue*, para iniciar la pausa, la expulsión del problemático y la terminación de la conexión.

8.3 Mantenimiento QoS

El mantenimiento QoS se ejecuta para mantener la calidad de la conexión en el nivel deseado y para evitar que la calidad de la conexión se degrade más allá del nivel de QoS negociado.

Basándose en los valores de estado de parámetro supervisados, el emisor puede tomar las medidas de mantenimiento QoS siguientes:

- 1) ajuste de la velocidad de transmisión de los datos,
- 2) pausa y reinicio de la conexión,
- 3) expulsión del problemático,
- 4) terminación de la conexión.

El ajuste de la velocidad de los datos está relacionado con el control del flujo en función de la velocidad y el control de la congestión. La pausa/reinicio y la terminación de la conexión son las medidas que se pueden tomar para gestionar la conexión. El emisor comunica estos eventos a todos los receptores mediante paquetes ND y CT.

Para iniciar estas medidas de mantenimiento QoS, el emisor tiene que configurar los valores umbral siguientes:

- 1) para el ajuste de la velocidad de la transmisión de datos:
threshold_rate_increase (incremento de umbral de velocidad) y *threshold_rate_decrease* (disminución de umbral de velocidad)
- 2) *threshold_connection_pause* (pausa de umbral de conexión)

Todos los valores umbral son números reales comprendidos entre 0 y 3.

8.3.1 Ajuste de la velocidad de transmisión de datos

El ECTP utiliza un control de flujo de tamaño fijo y basado en ventana. El parámetro *window size* (tamaño de ventana) por defecto es igual a *ACK Bitmap Size (ABS)* (tamaño de la correspondencia de bits de ACK): 32. El emisor puede transmitir como máximo los paquetes de datos *window size* a la velocidad de *Data TransmissionRate (DTR)* (velocidad de transmisión de datos). El ECTP efectúa el control de la congestión ajustando *DTR* dinámicamente sobre la base de los valores de estado del parámetro tasa de pérdida *Lvalue* (véase 8.2.4.2).

El ajuste de la velocidad de transmisión de los datos se basa en *threshold_rate_increase* y *threshold_rate_decrease*, que el emisor preconfigura a partir de los requisitos de la aplicación. Estos valores se colocan así:

$$0 \leq \text{threshold_rate_increase} \leq \text{threshold_rate_decrease} \leq 3.$$

Los valores por defecto son *threshold_rate_increase* = 1,0 y *threshold_rate_decrease* = 2,0.

En la fase de transmisión de los datos, el emisor comienza con $DTR = LQA \text{ throughput}$, y *DTR* se puede ajustar así:

$$LQA \text{ throughput} \leq DTR \leq CHQ \text{ throughput}.$$

Cada AGN segundos, el emisor ajusta *DTR* sobre la base de *threshold_rate_increase*, *threshold_rate_decrease* y *Lvalue* supervisado como sigue:

SI $Lvalue < threshold_rate_increase$,
 ENTONCES $DTR = \text{Min} \{CHQ, DTR + \text{Transmission Rate Increase (TRI)}\}$
 PERO SI $threshold_rate_increase \leq Lvalue \leq threshold_rate_decrease$
 ENTONCES *DTR* no se modifica
 PERO SI $Lvalue > threshold_rate_decrease$
 ENTONCES $DTR = \text{Max} \{LQA, DTR - \text{Transmission Rate Decrease (TRD)}\}$

Las variables de ajuste de velocidad como *TRI* y *TRD* (aumento o disminución de la velocidad de transmisión) se pueden establecer basándose en *CHQ* y *LQA* de caudal. Por ejemplo,

$$TRI = (CHQ - LQA) \times 1/20$$

$$TRD = (CHQ - LQA) \times 1/5$$

8.3.2 Pausa y reinicio de la conexión

El emisor puede hacer una pausa en la conexión para suspender temporalmente las transmisiones de datos multidifusión a fin de evitar que se degrade aún más la calidad de la conexión.

La pausa y el reinicio de la conexión se pueden llevar a cabo a petición de la aplicación. En este caso, la aplicación emisora inicia la pausa sobre la base de los valores de estado de parámetro supervisados *Tvalue*, *Dvalue*, *Jvalue* y *Lvalue*. Una vez iniciada la pausa, el emisor transmite periódicamente paquetes ND con el bit *F* fijado a '1' en la cabecera fija (véase la Rec. UIT-T X.606 | ISO/CEI 14476-1). El emisor no tiene que transmitir ningún paquete de datos DT nuevo mientras se envían los paquetes de control, incluidos los paquetes HB. Además, cada receptor puede enviar paquetes de control, como ACK.

La pausa también se puede iniciar sobre la base de *threshold_connection_pause*. En este caso, y únicamente si el valor de estado de conexión supervisada es mayor que *threshold_connection_pause*, el ECTP emisor inicia la pausa de conexión. Para *threshold_connection_pause* se propone el valor de 2,5.

En esta operación, el valor *estado de conexión* (Connection Status) se calcula para todos los valores de estado de parámetro supervisados como sigue.

$$Connection\ Status = Tweight \times Tvalue + Dweight \times Dvalue + Jweight \times Jvalue + Lweight \times Lvalue.$$

También hay que configurar cada uno de los valores ponderados, junto con *threshold_connection_pause*, para lo cual se imponen las siguientes restricciones:

$$0 \leq Tweight, Dweight, Jweight, Lweight \leq 1,$$

$$Tweight + Dweight + Jweight + Lweight = 1.$$

Un valor ponderado se fija a '0' si en la conexión no está habilitado el parámetro QoS correspondiente.

La pausa de conexión inicia cuando:

$$Connection\ Status \geq threshold_connection_pause.$$

Una vez indicada la pausa de conexión y si ha expirado el temporizador tiempo de pausa de conexión (*CPT*, *connection pause time*), se activa el reinicio de la conexión y el emisor empieza a transmitir datos multidifusión a la velocidad de transmisión de *LQA*. Cuando se indica el reinicio de la conexión, los paquetes ND fijan el bit *F* de la cabecera a '0'.

8.3.3 Expulsión del problemático

El emisor o LO pueden invocar la expulsión del problemático para mantener el estado de QoS en un nivel deseable y, además, para evitar que el estado de la conexión se degrade aún más. Las implementaciones pueden hacer un proceso diferente de expulsión del problemático sobre la base de los valores de estado de parámetro proporcionados en esta Especificación.

Por ejemplo, un receptor puede ser expulsado por su padre si ha comunicado un *parameter status value* más grande que *threshold connection pause* y varias veces mayor que un umbral preconfigurado. El diseño y la implementación del proceso de expulsión del problemático se han de hacer con precaución, ya que la operación de expulsión puede tener un impacto considerable en todo el comportamiento del protocolo ECTP.

8.3.4 Terminación de la conexión

La opción natural para terminación de la conexión es concluir la conexión una vez transmitidos todos los datos multidifusión. En las operaciones de gestión QoS también se inicia la terminación de la conexión cuando el estado de la conexión se percibe como "irrecuperable".

La terminación de la conexión se puede llevar a cabo a petición de la aplicación. Si se activa la terminación de la conexión, el emisor transmite un paquete CT a todos los receptores y cierra la conexión.

La terminación de la conexión también se puede activar sobre la base del temporizador *tiempo de terminación de conexión* (CTT, *connection termination time*) reconfigurado. En este caso, la terminación de la conexión se activa si

La pausa de conexión subsiguiente se vuelve a producir en el intervalo CTT a partir del reinicio de la conexión.

El temporizador CTT se activa cuando se indica el reinicio de la conexión. Algunas aplicaciones no soportan la terminación de la conexión.

9 Temporizadores y variables

A continuación, se indican los temporizadores y las variables que se utilizan en la gestión QoS.

9.1 Temporizadores

- a) Tiempo de generación de ACK (AGT, *AGK generation time*) en segundos: cada receptor genera periódicamente paquetes ACK cada AGT segundos, excepto para el primer paquete ACK. En ECTP-2, AGT se fija a AGN segundos (véase 8.2).
- b) Tiempo de pausa de conexión (CPT, *connection pause time*) en segundos: una vez indicada la pausa de conexión, la conexión hace una pausa durante el intervalo CPT (véase 8.3.2).
- c) Tiempo de terminación de la conexión (CTT) en segundos: una vez indicado el reinicio de la conexión, si se vuelve a producir la pausa de conexión dentro del intervalo CTT, se activa la terminación de la conexión (véase 8.3.4).
- d) Tiempo de supervisión de QoS (QMT, *QoS monitoring time*) en segundos: cada receptor mide los valores propios de los parámetros QoS durante cada intervalo QMT (véase 8.2.2).

9.2 Variables de funcionamiento

- a) Valores agregados de estado de parámetro: el emisor agrega los valores de estado de parámetro comunicados por los receptores, lo que arroja Tvalue, Dvalue, Jvalue, y Lvalue (véase 8.2).
- b) Número de generación ACK (AGN, *ACK generation number*): cada receptor genera paquetes ACK periódicamente cada AGT segundos. En ECTP-2, AGT se fija a AGN segundos (véase 8.2). Por defecto, AGN se fija a 8.
- c) Estado de conexión: para iniciar pausa/reinicio de la conexión, los valores agregados de estado de parámetro se pueden promediar por ponderación con los valores ponderados de parámetro preconfigurados Tweight, Dweight, Jweight, y Lweight, lo que arroja el parámetro *estado de la conexión*, que representa el estado completo de la conexión (véase 8.3.2).
- d) Velocidad de transmisión de datos (DTR, *data transmission rate*): el emisor transmite datos multidifusión a la velocidad DTR (véase 8.3.1).

- e) Valor medido de parámetro: cada receptor tiene que medir el valor propio de los parámetros QoS utilizados en la conexión, lo que arroja el *valor medido de parámetro* (véase 8.2).
- f) Valor de estado de parámetro: El *valor medido de parámetro* se hace corresponder con el *valor de estado de parámetro*, que es un entero como 0, 1, 2 ó 3 (véase 8.2).
- g) Disminución de la velocidad de transmisión (*TRD, transmission rate decrease*): *DTR* disminuye según *TRD* (véase 8.3.1).
- h) Aumento de la velocidad de transmisión (*TRI, transmission rate increase*): *DTR* aumenta según *TRI* (véase 8.3.1).
- i) *Threshold_rate_increase*: umbral para aumentar la velocidad de transmisión de datos (véase 8.3.1).
- j) *Threshold_rate_decrease*: umbral para disminuir la velocidad de transmisión de datos (véase 8.3.1).
- k) *Threshold_connection_pause*: umbral para activar la pausa de conexión (véase 8.3.2).

Anexo A

Interfuncionamiento de ECTP y RSVP para la reserva de recursos

(Este anexo no es parte integrante de esta Recomendación | Norma internacional)

Se puede garantizar la QoS de ECTP mediante el interfuncionamiento con RSVP/Intserv para los protocolos de reserva de recursos de red. Basándose en los valores de parámetro QoS, el protocolo RSVP puede configurar sus descriptores de tráfico si la función está habilitada en la red.

En este anexo se describe cómo utilizar la señalización RSVP junto con ECTP para la reserva de recursos y cómo se hace corresponder los parámetros QoS de ECTP con los descriptores de tráfico de RSVP. En todos los esquemas ilustrados a continuación se supone el modo sin negociación (véase 7.1 y 8.1), en el cual los valores de parámetro QoS propuestos por el emisor se imponen a todos los receptores sin negociación. Queda en estudio cómo utilizar RSVP en el modo con negociación. Sin embargo, aun en el modo con negociación, la relación de correspondencia de parámetros de RSVP y ECTP sigue siendo válida. En este caso, cada uno de los valores de parámetro QoS de ECTP "negociados" se hace corresponder con los descriptores de tráfico RSVP.

A.1 Parámetros QoS de ECTP

Los parámetros QoS de ECTP y sus valores objetivo se resumen así:

- a) caudal: CHQ, OT y LQA,
- b) retardo de tránsito: OT y LQA,
- c) fluctuación de fase del retardo de tránsito: OT y LQA,
- d) tasa de pérdida de datos: OT y LQA.

Para concluir, al menos desde el punto de vista de la tecnología RSVP actual el RSVP no puede soportar explícitamente el retardo de tránsito, la fluctuación de fase del retardo y la tasa de pérdida. La señalización RSVP sólo ofrece una garantía estricta del requisito de retardo en el nivel de red mediante la reserva de ancho de banda y el funcionamiento de un programador de tráfico en los enrutadores intermedios del trayecto. El significado de "retardo" en RSVP (retardo de puesta en cola en el nivel de red) es bastante distinto del que tiene en ECTP (retardo de tránsito de extremo a extremo). La subcláusula siguiente es un resumen de las funciones de RSVP.

A.2 Visión general de RSVP

El protocolo RSVP (RFC 2205, 2210, 2212 del IETF) incluye los siguientes objetos RSVP:

- a) RSVP SENDER_TSPEC,
- b) RSVP ADSPEC,
- c) RSVP RECEIVER_FLOWSPEC.

A.2.1 RSVP SENDER_TSPEC

La especificación de tráfico (TSPEC) de RSVP es un objeto que transporta los descriptores de tráfico generados por un emisor en una sesión RSVP. Este objeto consiste en los descriptores de tráfico siguientes:

- a) especificación de los cubos de testigos r y b , donde r es la generación de testigos o tasa de fuga, y b es la profundidad del cubo,
- b) velocidad de cresta p ,
- c) unidad mínima vigilada m ,
- d) tamaño máximo de paquete M .

Los descriptores r y p se miden en bytes por segundo, y b , m y M se miden en bytes. Las velocidades r y p son las velocidades media y máxima del flujo de información, respectivamente, y b es un parámetro que acota la variabilidad de la emisión de tráfico.

El resto de los parámetros establece límites para la distribución de la longitud del paquete dentro del flujo. M es el tamaño máximo de un paquete conforme, y m es tal que el vigilante de la red trata cualquier paquete de tamaño menor que m como si tuviera un tamaño igual a m . Los descriptores m y M no participan directamente en el ECTP.

A.2.2 ADSPEC de RSVP

La especificación de anuncio (ADSPEC, *advertisement specification*) de RSVP es un objeto que transporta información generada en los emisores de datos o en los elementos de red intermedios. La ADSPEC de RSVP fluye en sentido descendente hacia los receptores, y se puede utilizar y actualizar dentro de la red antes de su entrega a las aplicaciones receptoras. Esta información incluye tanto los parámetros que describen las propiedades del trayecto de datos, como los relativos a la disponibilidad de los servicios de control QoS específicos, como los parámetros que necesitan los servicios de control QoS específicos para funcionar correctamente. Este objeto incluye los descriptores de tráfico siguientes:

- Ancho de banda del enlace con el ancho de banda mínimo disponible en el trayecto de extremo a extremo.
- Retardos de transmisión extremo a extremo, excepto el retardo de puesta en cola; estos retardos se pueden medir mediante dos términos de error, C y D . El término de error C depende de la velocidad. Representa el retardo que podría sufrir un paquete en el flujo debido a los parámetros de velocidad del flujo. El término de error D es independiente de la velocidad, elemento por elemento. Representa la variación temporal del tránsito no basada en la velocidad para el caso más desfavorable a través del elemento de servicio. Generalmente se determina o se fija en el momento de arranque o de configuración.

A.2.3 RSVP FLOWSPEC

Este objeto transporta la información de petición reserva generada por los receptores, y consiste en RECEIVER_TSPEC y RECEIVER_RSPEC. La información en FLOWSPEC fluye en sentido ascendente de los receptores de datos hacia las fuentes de datos. Esta información se puede utilizar o actualizar en los elementos de red intermedios antes de llegar a la aplicación emisora.

RECEIVER_TSPEC es idéntica a SENDER_TSPEC, salvo en lo que hace a la MTU del trayecto extremo a extremo. Por otro lado, RECEIVER_RSPEC incluye los parámetros siguientes:

- R : ancho de banda reservado,
- S : término de holgura.

R ha de ser igual o superior a r de TSPEC. El término de holgura S está dado en microsegundos y significa la diferencia entre el retardo deseado y el retardo obtenido al utilizar el nivel de reserva R . El elemento de red puede utilizarlo para reducir su reserva de recursos correspondiente al flujo asociado.

A.2.4 RSVP API

En la figura A.1 se ilustra la estructura de implementación de RSVP en un sistema.

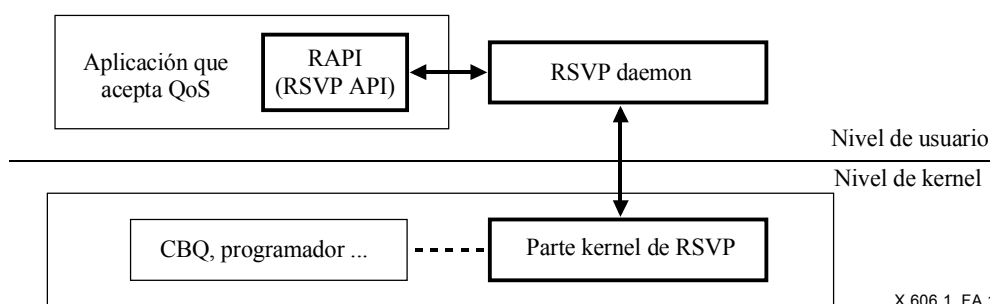


Figura A.1 – Estructura de implementación de RSVP

En un sistema, el protocolo RSVP funciona con tres módulos: kernel, daemon y RAPI. El kernel maneja la puesta en cola y la planificación de los datos de conformidad con los requisitos del RSVP de la capa superior. El daemon maneja la señalización entre RAPI y kernel. La RAPI proporciona una interfaz con las aplicaciones que aceptan QoS, lo que garantiza que las operaciones de RSVP soportan los requisitos de QoS de las aplicaciones. Las características QoS de ECTP se integran en las redes con capacidad RSVP y la RAPI las aplica en las redes que trabajan con RSVP.

Las funciones RAPI se clasifican en dos grupos: las funciones invocadas por una aplicación y las rutinas asíncronas (llamada ascendente, *upcall*) entregadas a una aplicación mediante el daemon RASP.

Estas funciones RSVP básicas son:

- rapi_session(), se utiliza para establecer una sesión RSVP;
- rapi_sender(), la utiliza el emisor para enviar mensajes RSVP PATH;
- rapi_reserve(), la utiliza el receptor para reservar recursos de red;
- rapi_release(), se utiliza para terminar una sesión RSVP;
- rapi_getfd() y rapi_dispatch(), las utiliza la aplicación para obtener información de estado con respecto a RSVP a través de las rutinas upcall asíncronas.

A.3 Ejemplo de correspondencia de parámetros RSVP con parámetros ECTP

El cuadro A.1 siguiente es un resumen de ejemplo de correspondencia de parámetros RSVP con parámetros ECTP, que se puede utilizar para dar soporte QoS junto con RSVP.

NOTA – Según la implementación se puede utilizar una correspondencia distinta.

Cuadro A.1 – Correspondencia de parámetros RSVP y ECTP

Parámetros RSVP	Parámetro QoS de ECTP	Descripción
p (TSPEC)	Caudal CHQ	Correspondencia directamente con p
r (TSPEC)	Caudal OT	Correspondencia directamente con r
b (TSPEC)	(Caudal CHQ – Caudal OT) $\times$ un tiempo arbitrario (1 ~ 3 segundos)	El tamaño del cubo de testigos se fija arbitrariamente basándose en los parámetros caudal de ECTP.
m (TSPEC)	Cabecera fija IP+UDP+ECTP (20 + 8 + 16 = 44 bytes)	Correspondencia directa
M (TSPEC)	MSS ECTP (1024 bytes por defecto)	Correspondencia directa
R (RSPEC)	(Caudal CHQ + caudal OT) / 2	La reserva se fija al valor medio de los caudales CHQ y OT
S (RSPEC)	0	No se fija el término holgado

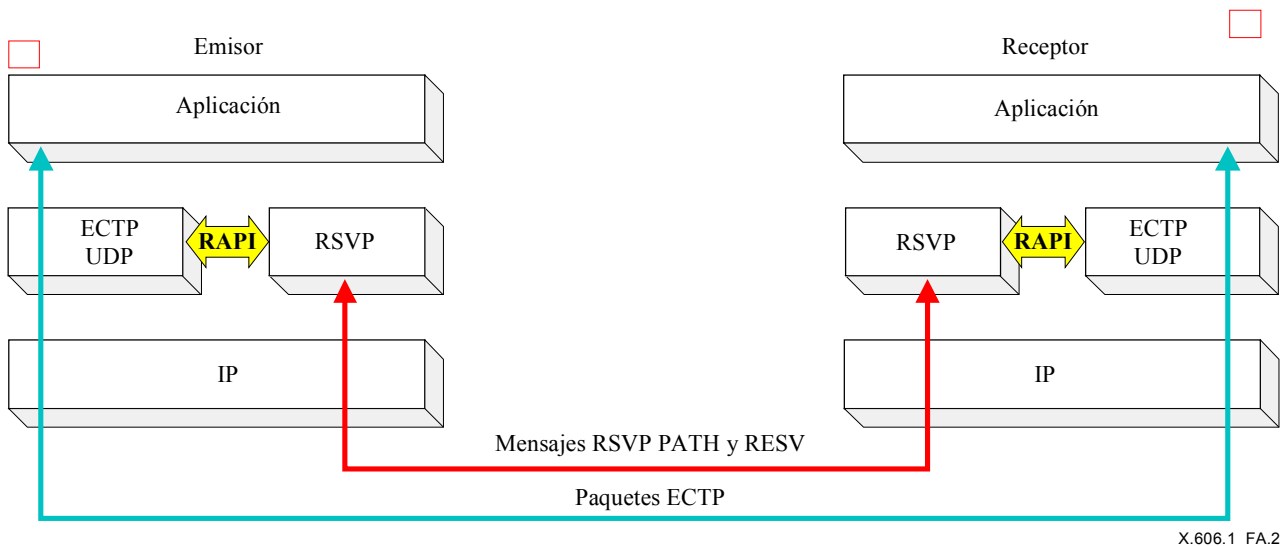
Cuando el daemon de ECTP invoca al daemon de RSVP, los parámetros QoS ECTP se hacen corresponder con los descriptores de tráfico RSVP TSPEC: p, r, b, m, M, como se ilustra en el cuadro. Estos valores de parámetro TSPEC se transmiten del emisor a los receptores mediante mensajes RSVP PATH.

Cuando llega el mensaje RSVP PATH a un receptor, la información de los objetos SENDER_TSPEC y ADSPEC pasa a través de RSVP API hasta la aplicación ECTP. La aplicación interpreta la información recibida y la utiliza para seleccionar los parámetros de reserva de recursos. Estos parámetros se componen en un objeto RSVP FLOWSPEC que se transmite al emisor mediante mensajes RSVP RESV. La reserva de ancho de banda R y el término de holgura S , contenidos en el objeto RSVP RSPEC, se hacen corresponder con los parámetros ECTP como se ilustra en el cuadro A.1.

A.4 Caso de interfuncionamiento de ECTP y RSVP

Los valores objetivo de cada parámetro QoS ECTP se pueden utilizar para construir un objeto RSVP SENDER_TSPEC. Se necesita una interfaz entre ECTP y RSVP para comunicar los valores de parámetro QoS ECTP al procesador RSVP. El objeto SENDER_TSPEC se transmite a los receptores mediante mensajes RSVP PATH.

El momento en el que ECTP llama al RSVP depende de la implementación, pero se recomienda que el emisor ECTP invoque al RSVP después de transmitir el paquete CR y que el receptor lo haga una vez recibido ese mensaje. En la figura A.2 se ilustra un modelo de interfuncionamiento de ECTP y RSVP.



X.606.1_FA.2

Figura A.2 – Modelo de interfuncionamiento de ECTP y RSVP

Como se muestra en la figura, las comunicaciones entre ECTP y RSVP se efectúan mediante RSVP API (RAPI). El emisor ECTP transfiere la información sobre los parámetros QoS al RSVP mediante la RAPI, cuando invoca al daemon de RSVP. Una vez que el ECTP activa el RSVP, se establecen las comunicaciones RSVP entre el daemon RSVP emisor y los daemon RSVP receptores. O sea que entre el emisor y los receptores se establece una relación de entidades par RSVP.

Durante la conexión ECTP, cada módulo RSVP comunica el estado de la reserva de recursos de red a su daemon ECTP asociado por intermedio de la RAPI, lo que permite conocer si la reserva hizo satisfactoriamente o en condiciones de error. Un ejemplo de un conjunto de códigos de estado RSVP es:

- Código de estado 0: Se indican eventos PATH (utilizados por el RSVP receptor)
- Código de estado 1: Se indican eventos de error PATH (utilizados por el RSVP emisor)
- Código de estado 2: Se indican eventos RESV (utilizados por el RSVP emisor)
- Código de estado 3: Se indican eventos de error RESV (utilizados por el RSVP receptor)
- Código de estado 4: Se indican eventos RESV_CONFIRM (utilizados por el RSVP receptor)

El núcleo del protocolo ECTP puede obtener estos códigos de estado mediante las funciones *upcall* asíncronas RSVP del daemon RSVP.

Los procedimientos que realizan el ECTP y los módulos RSVP para el interfuncionamiento se resumen así:

- (1) El emisor ECTP transmite el paquete CR a los receptores. El paquete CR contiene el elemento QoS que indica los valores de parámetro QoS que necesita la aplicación.
- (2) El emisor ECTP invoca a su daemon RSVP asociado.
- (3) El emisor RSVP prepara el mensaje PATH, que contiene los parámetros TSPEC basados en los parámetros QoS ECTP.
- (4) El emisor RSVP transmite periódicamente los mensajes PATH a los receptores.
- (5) Si se indica un error RSVP PATH, el emisor RSVP comunica el código de estado correspondiente al emisor ECTP.
- (6) No bien el receptor ECTP recibe un paquete CR, invoca al daemon del receptor RSVP asociado.
- (7) Fundándose en el mensaje PATH recibido del emisor, el receptor RSVP prepara los mensajes RESV correspondientes, incluidos los parámetros RSPEC. Se puede configurar una parte de la información de RSPEC, como *R* (ancho de banda) y *S* (término de holgura), mediante la interacción del receptor RSVP y el receptor ECTP.
- (8) El receptor RSVP transmite los mensajes RESV correspondientes al emisor.
- (9) Cuando se indica un error RESV, el receptor RSVP comunica el código de estado correspondiente al receptor ECTP.

Cuando ECTP recibe un código de estado de su RSVP asociado, puede indicar la información a su aplicación. El uso detallado de los códigos de estado depende de las implementaciones.

En la figura A.3 se ilustran los flujos de mensaje ECTP y RSVP entre el emisor y los receptores. Después de transmitir un paquete HB, el emisor ECTP puede invocar un mensaje RSVP PATH inicial. Los mensajes RSVP RESV correspondientes vienen de los receptores ECTP. Los mensajes RSVP PATH subsiguientes se inician y repiten junto con los paquetes HB ECTP periódicos.

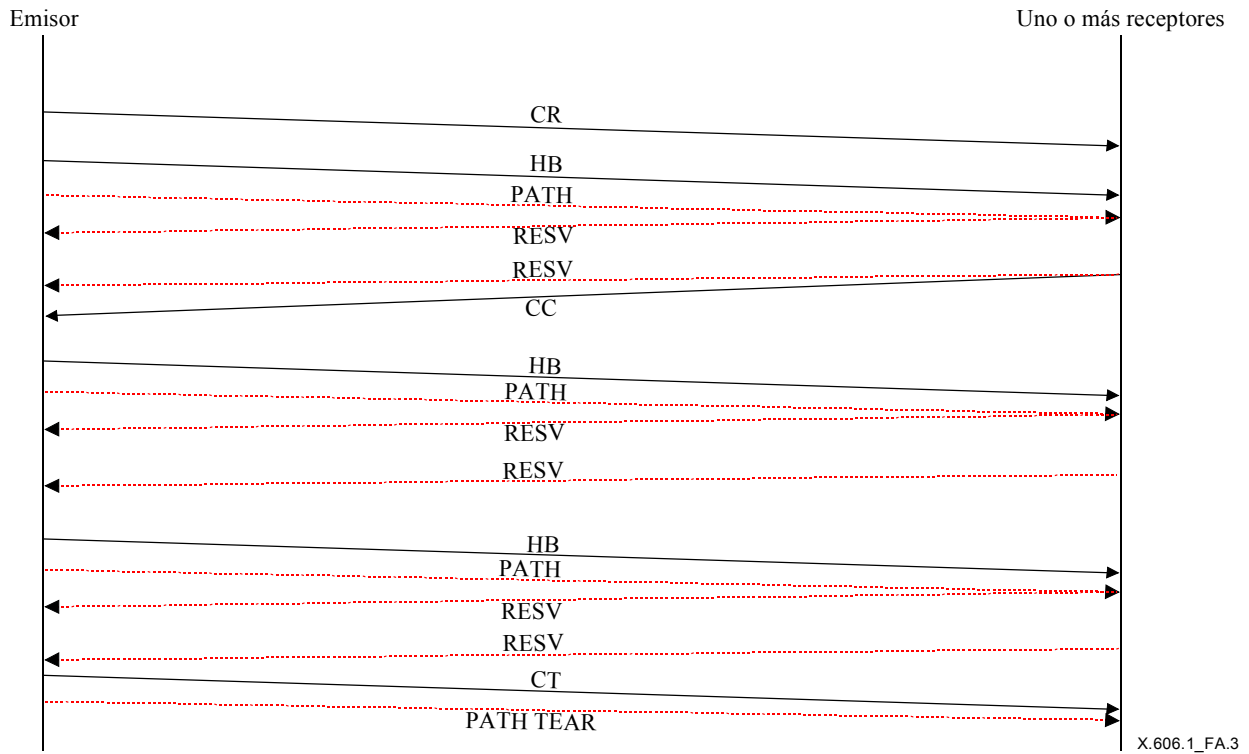


Figura A.3 – Modelo de interfuncionamiento de ECTP y RSVP

Anexo B

Interfaces de programación de aplicación

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

En este anexo se especifican las interfaces de programación de aplicación (API) para ECTP. La ECTP API descrita en esta Recomendación | Norma Internacional puede ser utilizada por aplicaciones que emplean las capacidades de transporte de la parte 1 de ECTP (Rec. UIT-T X.606 | ISO/CEI 14476-1) y la parte 2 de ECTP (Rec. UIT-T X.606.1 | ISO/CEI 14476-2).

Esta API se diseñó basándose en las funciones API de zócalo Berkeley. Sin embargo, para distinguir la ECTP API de las funciones de zócalo Berkeley existentes, las funciones ECTP API llevan el prefijo "*m*" (por ejemplo, `msocket`).

B.1 Visión general

B.1.1 Funciones API

En el cuadro B.1 se resumen las funciones API utilizadas en ECTP.

Cuadro B.1 – Funciones ECTP API

Nombre de función	Descripción
<code>msocket()</code>	Crea un nuevo zócalo multidifusión en el dominio de comunicación ECTP.
<code>Mbind()</code>	Asocia un conjunto de direcciones/puertos locales y de grupo con el zócalo.
<code>Mconnect()</code>	El emisor inicia la creación de una conexión a una dirección exterior específica. El receptor tardío inicia el proceso de incorporación.
<code>maccept()</code>	Los posibles receptores se unen a la conexión STP al aceptar la señal de creación de conexión del emisor.
<code>Msend()</code>	Envía datos de aplicación a un grupo de destino.
<code>Mrecv()</code>	Entrega los datos recibidos a la aplicación. Entrega algunos mensajes de indicación para el control de la aplicación durante la fase de transferencia de datos
<code>mclose()</code>	Termina la conexión y libera el zócalo.
<code>mgetsockopt()</code>	Obtiene las opciones de zócalo y de protocolo del kernel.
<code>msetsockopt()</code>	Establece las opciones de zócalo y de protocolo hacia el kernel.

B.1.2 Utilización de las funciones ECTP API

En la figura B.1 se ilustra un ejemplo de utilización de las funciones ECTP API. Las secuencias de las funciones API invocadas se indican en términos de emisor, receptor y receptor tardío.

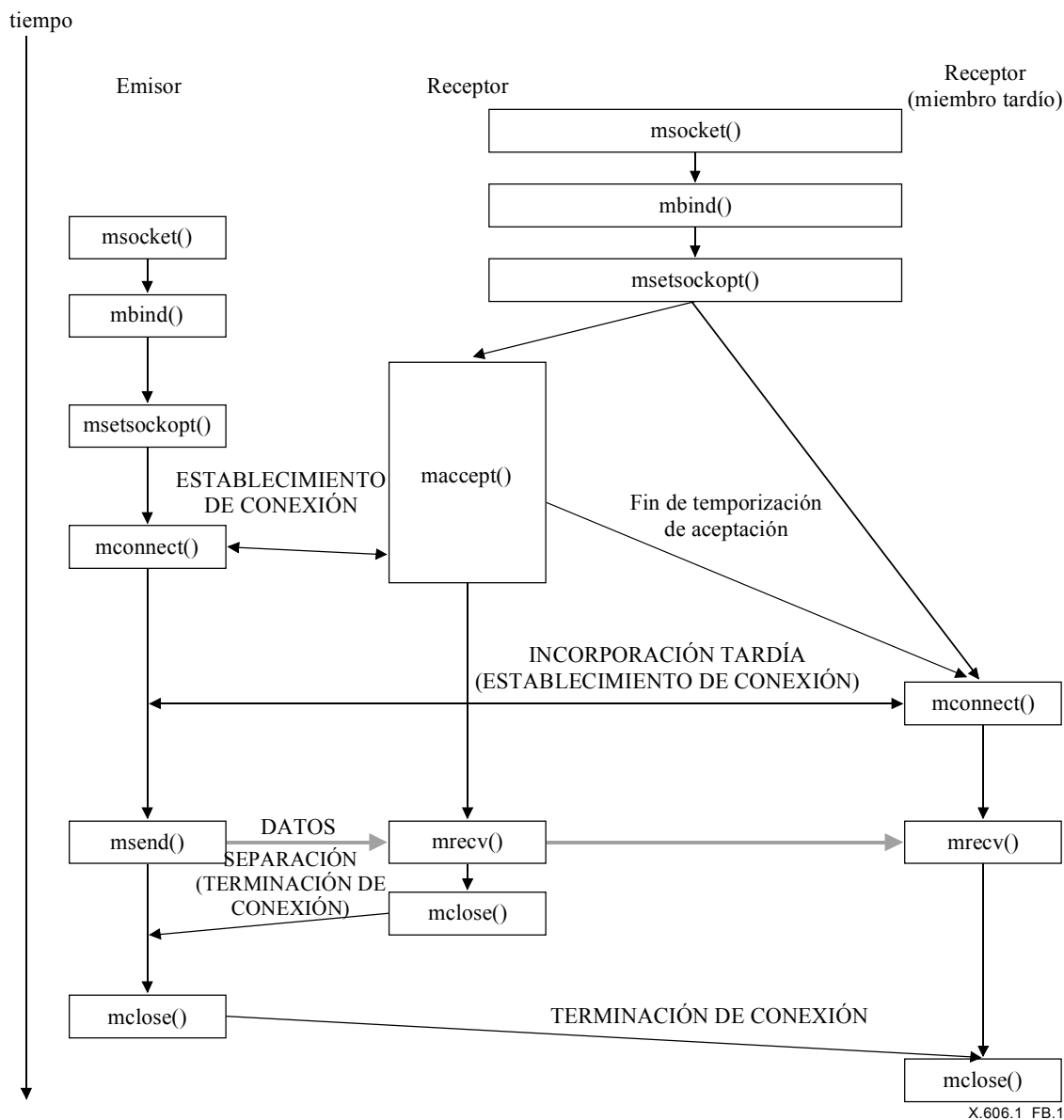


Figura B.1 – Utilización de ECTP API

Como se indica en la figura, el receptor toma distintas medidas según su tipo de nodo: un receptor que se incorpora anticipadamente o uno que se incorpora tardíamente. El receptor que se incorpora anticipadamente pasa al modo `maccept` después de `msetsockopt`, mientras que el receptor que se incorpora tardíamente invoca la función `mconnect` directamente. El receptor que se incorpora anticipadamente puede pasar al modo `mconnect` después de una duración de fin de temporización de aceptación específica, si no puede recibir ninguna señal de indicación de creación de conexión.

B.2 Funciones ECTP API

B.2.1 `msocket()`

Para utilizar ECTP, una aplicación TIENE QUE invocar primero la función `msocket`, que especifica el tipo de protocolo de comunicación deseado, como ECTP con IPv4, ECTP con IPv6, ECTP con protocolos OSI, etc.

```
int msocket(int family, int type, int protocol);
```

Descripción de parámetros:

- *familia*: especifica la familia de protocolos y es una de las constantes indicadas en el cuadro B.2;
- *tipo*: especifica el tipo de zócalo y es una de las constantes indicadas en el cuadro B.3;
- *protocolo*: se fija a cero.

Cuadro B.2 – Constantes de *familia* de protocolos utilizadas para la función *msocket*

familia	Descripción
AF_INET	Protocolos IPv4
AF_INET6	Protocolos IPv6
AF_ISO	Protocolos de dominio OSI

Cuadro B.3 – *Tipo* de zócalo utilizado para la función *msocket*

Tipo	Descripción
SOCK_ECTP1	Zócalo para la parte 1 de ECTP
SOCK_ECTP2	Zócalo para la parte 2 de ECTP

En el cuadro B.4 se indican las combinaciones válidas, junto con el protocolo real seleccionado por el par:

Cuadro B.4 – Combinaciones de *familia* y *tipo* para la función *msocket*

Tipo	AF_INET	AF_INET6	AF_ISO
SOCK_ECTP1	ECTP1	ECTP1	ECTP1
SOCK_ECTP2	ECTP2	ECTP2	ECTP2

La llamada *msocket* devuelve un descriptor no negativo si tiene éxito, o -1 en los siguientes casos relacionados en el cuadro B.5.

Cuadro B.5 – Códigos de error de la llamada *msocket*

Código de error	Descripción
EPROTONOSUPPORT	Dentro de este dominio no se soporta el tipo de protocolo o el protocolo especificado.
EMFILE	El cuadro del descriptor por proceso está lleno.
ENFILE	El cuadro de fichero de sistema está lleno.
EACCES	Se niega el permiso para crear un zócalo del tipo y/o protocolo especificado.
ENOBUFS	No hay suficiente espacio en la memoria tampón.

B.2.2 *mbind()*

La función *mbind* asigna un conjunto de direcciones de control locales o de grupo, y el rol del nodo en la sesión hacia un zócalo. En los protocolos Internet, la dirección de protocolo es la combinación de una dirección IPv4 de 32 bits o una dirección IPv6 de 128 bits, junto con un número de puerto de 16 bits.

```
int mbind(int msockfd, const struct sockaddr *laddr, socklen_t laddrlen, const struct
sockaddr *gaddr, socklen_t gaddrlen, struct sockaddr *caddr, socklen_t caddrlen, int
role );
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función *msocket*;
- *laddr*: puntero a una dirección específica de protocolo para vincular una dirección local con el zócalo antes indicado;
- *laddrlen*: tamaño de la estructura de dirección indicada más arriba;
- *gaddr*: puntero a una dirección específica de protocolo para vincular una dirección de grupo objetivo con el zócalo;
- *gaddrlen*: tamaño de la estructura de dirección de grupo.
- *caddr*: puntero a una dirección específica de protocolo para vincular una dirección de control con el zócalo;
- *caddrlen*: tamaño de la estructura de dirección de control; y
- *role*: especifica el rol de este iniciador llamante, como TO, LO o LE.

Cuadro B.6 – Rol del usuario del zócalo para la función *msocket*

Rol	Descripción
TO	Creador de la conexión y emisor en tanto que propietario en las comunicaciones ECTP
LO	Receptor que asume la responsabilidad de las retransmisiones en la jerarquía basada en árbol
LE	Receptor no designado como LO

Una aplicación puede aplicar una función *mbind* a una dirección IP específica y una dirección de red de grupo a su zócalo. Las direcciones de grupo y de fuente deben pertenecer a una interfaz en el *host*.

La llamada *mbind* devuelve cero si tiene éxito o -1 según los motivos enumerados en el cuadro B.7.

Cuadro B.7 – Códigos de error que puede provocar *mbind*

Código de error	Descripción
EAGAIN	Los recursos de kernel para completar la petición están indisponibles temporalmente.
EBADF	<i>msockfd</i> no es un descriptor válido.
ENOTSOCK	<i>msockfd</i> no es un zócalo.
EADDRNOTAVAIL	La dirección especificada no está disponible en la máquina local.
EADDRINUSE	La dirección especificada ya está en uso.
EACCES	La dirección solicitada está protegida, y el usuario actual no tiene el permiso adecuado para el acceso.
EFAULT	El parámetro de dirección no está en una parte válida del espacio de dirección de usuario.
EROLE	El rol solicitado no es válido.
NOTA – El código de error sombreado se define por primera vez para ECTP.	

B.2.3 *maccept()*

Únicamente un miembro de sesión pasivo, como LE o LO, puede invocar esta función. Puede esperar que el emisor inicie un fin de temporización específico mediante *msetsockopt* y comunica si la conexión multidifusión se estableció o no.

```
int maccept(int msockfd, struct sockaddr *raddr, socklen_t *raddrlen);
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función *msocket*;
- *raddr*: devuelve la dirección de protocolo del iniciador de conexión distante (el emisor o TO); y
- *raddrlen*: puntero al tamaño de la estructura de dirección de zócalo señalada por *raddr*.

Si *maccept* tiene éxito, devuelve el mismo valor que el primer argumento, *msockfd*. Por ello, al valor devuelto se le denomina descriptor de zócalo *conectado*.

La llamada `maccept` devuelve un descriptor no negativo si tiene éxito, o `-1` por los motivos enumerados en el cuadro B.8.

Cuadro B.8 – Códigos de error utilizados para `maccept`

Código de error	Descripción
EBADF	El descriptor no es válido.
EINTR	Se interrumpió la operación <code>maccept</code> .
EMFILE	El cuadro de descriptor por proceso está lleno.
ENFILE	El cuadro de fichero de sistema está lleno.
ENOTSOCK	El descriptor hace referencia a un fichero, no a un zócalo.
EFAULT	El parámetro <i>addr</i> no está en la parte en la que se puede escribir del espacio de dirección de usuario.
EWOULDBLOCK	El zócalo está marcado sin bloqueo y no hay conexiones pendientes de aceptación.
ECONNABORTED	Llegó una conexión, pero se la cerró mientras esperaba en la cola de escucha.
ECRTIMEOUT	Indica que expiró el tiempo de espera de CR.
NOTA – El código de error sombreado se define por primera vez en ECTP.	

B.2.4 `mconnect()`

El TO o el LE incorporado tardíamente utilizan la función `mconnect` para establecer una conexión.

```
int mconnect(int msockfd, const struct sockaddr *daddr, socklen_t daddrlen);
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función `msocket`;
- *daddr*: puntero a una dirección de destino específica de protocolo. Puede ser una dirección de grupo (cuando el iniciador llamante es TO), o una dirección de emisor (cuando el iniciador llamante es LO o LE);
- *daddrlen*: tamaño de *daddr*.

`mconnect` devuelve cero si tiene éxito o `-1` en los casos anormales enumerados en el cuadro B.9.

Cuadro B.9 – Números de error de la función `mconnect`

Código de error	Descripción
EBADF	<i>msockfd</i> no es un descriptor válido.
ENOTSOCK	<i>msockfd</i> es un descriptor de fichero, no de zócalo.
EADDRNOTAVAIL	En esta máquina no está disponible la dirección especificada.
EAFNOSUPPORT	Las direcciones de la familia de direcciones especificada no se pueden utilizar con este zócalo.
EISCONN	El zócalo ya está conectado.
ECONNREFUSED	El intento de conectar fue rechazado por fuerza.
ENETUNREACH	No se puede llegar a la red desde este <i>host</i> .
EADDRINUSE	La dirección ya está en uso.
EFAULT	El parámetro de nombre especifica una zona fuera del espacio de dirección de proceso.
EALREADY	El zócalo es sin bloqueo y no se ha completado aún el intento de conexión previo.
EDENIED	Indica que el TO no aceptó la petición incorporación de LE o LO.
ETIMEDOUT	Expiró el establecimiento de la conexión sin lograr establecerla. Indica que no hay respuesta del TO.
NOTA – Los códigos de error sombreados se definen por primera vez en ECTP.	

B.2.5 msend()

Esta función `msend` escribe datos de una memoria tampón en un zócalo conectado.

```
ssize_t msend (int msockfd, const void *buf, size_t buflen, int *flags);
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función `msocket`;
- *buf*: puntero a una memoria tampón para escribir;
- *buflen*: tamaño de *buf*, y
- *flags*: aún no se ha definido.

`msend` devuelve el número de bytes escritos si tuvo éxito o `-1` en los casos anormales enumerados en el cuadro B.10.

Cuadro B.10 – Códigos de error de `msend`

Código de error	Descripción
EBADF	Se especificó un descriptor no válido.
EACCES	La dirección de destino es una dirección de radiodifusión, y no se ha fijado <code>SO_BROADCAST</code> en el zócalo.
ENOTSOCK	El argumento <i>msockfd</i> no es un zócalo.
EFAULT	Se especificó una dirección de espacio de usuario no válida para un parámetro.
EMSGSIZE	El zócalo necesita que el mensaje se envíe atómicamente, y el tamaño del mensaje que hay que enviar no lo permite.
EAGAIN	El zócalo está marcado sin bloqueo y la operación solicitada se bloqueará.
ENOBUFS	El sistema no pudo asignar una memoria tampón interna. La operación se podrá realizar haya memorias tampón disponibles.
ENOBUFS	La cola de salida de una interfaz de red estaba llena. Esto indica normalmente que la interfaz ya no está transmitiendo, pero puede deberse a una congestión pasajera.
EPARTITIONED	Indica que la sesión se ha dividido.
NOTA – El código de error sombreado se define por primera vez en ECTP.	

B.2.6 mrecv()

La función `mrecv` se utiliza para recibir los datos multidifusión y las señales de indicación a efectos de control.

```
ssize_t mrecv (int msockfd, void *buf, size_t buflen, int *flags, struct sockaddr *fromaddr, socklen_t *fromaddrlen);
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función `msocket`;
- *buf*: puntero a una memoria tampón para leer algo dentro;
- *buflen*: tamaño de *buf*;
- *flags*: aún no se ha definido;
- *fromaddr*: puntero a una dirección específica de protocolo que determina el emisor;
- *fromaddrlen*: tamaño de *fromaddr*.

Cuando una aplicación recibe datos de la memoria tampón, puede identificar el emisor correspondiente mediante *fromaddr*.

La función `mrecv` devuelve el número de bytes recibidos si tuvo éxito. De lo contrario devuelve `-1` si se produce un error o si hay un mensaje de control que se entregará a la aplicación. Los códigos de error se enumeran en el cuadro B.11.

Cuadro B.11 – Códigos de error de la función `mrecv`

Código de error	Descripción
EBADF	El argumento <i>msockfd</i> es un descriptor no válido.
ENOTCONN	El zócalo está asociado con un protocolo orientado a conexión y no se ha conectado (véase <code>mconnect</code> y <code>maccept</code>).
ENOTSOCK	El argumento <i>msockfd</i> no se refiere a un zócalo.
EAGAIN	El zócalo está marcado sin bloqueo, y la operación de recepción se bloqueará, o se ha establecido un límite de temporización de recepción, que expiró antes de que los datos se hubieran recibido.
EINTR	La recepción se interrumpió con la llegada de una señal antes de que estuvieran disponibles los datos.
EFAULT	El puntero(s) a la memoria tampón de recepción apunta fuera del espacio de dirección del proceso.
ETOTERM	TO dio por terminada la sesión.
ETOEPEL	TO expulsó a LE o LO.
EPARTITIONED	Indica que la sesión se ha dividido.
NOTA – Los códigos de error sombreados se definen por primera vez en ECTP.	

B.2.7 `mclose()`

La función `mclose` se utiliza para abandonar o terminar una conexión ECTP mediante el cierre del zócalo. La acción por defecto de `mclose` con un zócalo ECTP es marcar el zócalo como cerrado y regresar al proceso inmediatamente. El proceso no puede seguir usando el descriptor de zócalo.

```
int mclose (int msockfd);
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función `msocket`.

`mclose` devuelve cero si tuvo éxito o -1 en el caso de los errores enumerados en el cuadro B.12.

Cuadro B.12 – Códigos de error de `mclose`

Código de error	Descripción
EBADF	<i>msockfd</i> no es un descriptor activo.
EINTR	Se recibió una interrupción.

B.2.8 `mgetsockopt()` y `msetsockopt()`

Se utiliza `mgetsockopt` para obtener las opciones y las características de conexión actuales que afectan a un zócalo.

```
int mgetsockopt(int msockfd, int level, int optname, void *optval, socklen_t *optlen);
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función `msocket`;
- *level*: especifica el código en el sistema para interpretar la opción: el código de zócalo general, o algún código específico de protocolo (por ejemplo, IPv4, IPv6, o ECTP);
- *optname*: especifica el tipo de opción. Cada nivel puede definir varios nombres de opción;
- *optval*: puntero a una variable en la que `mgetsockopt` almacena el valor actual de la opción; y
- *optlen*: especifica el tamaño de *optval* como un resultado de valor para `mgetsockopt`.

La función `msetsockopt` se utiliza para fijar las opciones que afectan a un zócalo.

```
int msetsockopt(int msockfd, int level, int optname, const void *optval, socklen_t *optlen);
```

Descripción de parámetros:

- *msockfd*: descriptor de zócalo devuelto por la función `msocket`;
- *level*: especifica el código en el sistema para interpretar la opción: el código de zócalo general o algún código específico de protocolo (por ejemplo, IPv4, IPv6, o ECTP);
- *optname*: especifica el tipo de opción. Cada nivel puede definir varios nombres de opción;
- *optval*: puntero a una variable desde la cual se recoge el nuevo valor de la opción mediante `msetsockopt`; y
- *optlen*: especifica el tamaño de *optval* como un valor para `msetsockopt`.

En el cuadro B.13 se resumen las opciones que pueden ser indagadas por `mgetsockopt` y `msetsockopt`.

Cuadro B.13 – Opciones de zócalo para `mgetsockopt` y `msetsockopt` con ECTP

Nivel	optname	mget	mset	Descripción	Bandera	Datatype
IPPROTO_ECTP	ECTP_QOS	•	•	Obtener y fijar los parámetros QoS		QoS
	ECTP_OPPAR		•	Obtener y fijar la dirección del padre de control		OP_par
	ECTP_OPVAR1	•	•	Obtener y fijar los valores de funcionamiento para la parte I de ECTP		OP_var1
	ECTP_OPVAR2	•	•	Obtener y fijar los valores de funcionamiento para la parte II de ECTP		OP_var2
	ECTP_OPLJ	•	•	Habilita la incorporación tardía	•	int
	ECTP_OPWCR	•	•	Fija el tiempo de espera de CR		u_long

`mgetsockopt` y `msetsockopt` devuelven cero si tuvieron éxito o `-1` si ocurren los errores enumerados en el cuadro B.14.

Cuadro B.14 – Códigos de error para `mgetsockopt` y `msetsockopt`

Código de error	Descripción
EBADF	El argumento <i>msockfd</i> no es un descriptor válido.
ENOTSOCK	El argumento <i>msockfd</i> es un fichero, no un zócalo.
ENOPROTOOPT	La opción es desconocida en el nivel indicado.
EFAULT	La dirección señalada por <i>optval</i> no está en una parte válida del espacio de dirección del proceso. En el caso de <code>mgetsockopt</code> , también se puede devolver este error si <i>optlen</i> no está en una parte válida del espacio de dirección del proceso.

Las siguientes estructuras de datos se ilustran como información para uso de `mgetsockopt` o `msetsockopt`.

```
typedef struct _op_var1 {
    int agn; /* ACK Generation Number */
    int mcn; /* Maximum Children Number */
    int mrn; /* Maximum Retransmission Number */
    int mtl; /* Maximum Tree Level */
    int nft; /* Node Failure Threshold */
    int arn; /* Active Receiver Number */
    int ccn; /* Current Children Number */
    int crn; /* Current Receiver Number */
    int ctl; /* Current Tree Level */
    int ctr; /* Current Transmission Rate */
    int cct; /* Connection Creation Time */
}OP_var1;
```

```
typedef struct _op_var2 {
    int mintr; /* Minimum Transmission Rate */
    int ctr; /* Current Transmission Rate */
    int maxtr; /* Maximum Transmission Rate */
}OP_var2;
```

```
typedef struct _op_par {
    u_long ctladdr; /* Local Group Control Multicast Address */
    u_short ctlport; /* Local Group Control Port */
}OP_par;
```

```
typedef struct _QoS {
    u_short threshold; /* threshold rate */
    u_long th_CHQ_TQA:2, th_CHQ_N:1, th_CHQ:29; /* CHQ throughput */
    u_long th_OT_TQA:2, th_OT_N:1, th_OT:29; /* OT throughput */
    u_long th_LQA_TQA:2, th_LQA_N:1, th_LQA:29; /* LQA throughput */
    u_long td_OT_TQA:2, td_OT_N:1, td_OT:29; /* OT transit delay */
    u_long td_LQA_TQA:2, td_LQA_N:1, td_LA:29; /* LQA transit delay */
    u_long tdj_OT_TQA:2, tdj_OT_N:1, tdj_OT:29; /* OT transit delay jitter */
    u_long tdj_LQA_TQA:2, tdj_LQA_N:1, tdj_LQA:29; /* LQA transit delay jitter */
    u_long cr_TSDU_TQA:2, cr_TSDU_N:1, cr_TSDU:29; /* corrupted data rate */
    u_long l_TSDU_TQA:2, l_TSDU_N:1, l_TSDU:29; /* lost data rate */
}QoS;
```

B.3 Ejemplo del fichero de encabezamiento msocket.h

A continuación se describe a título informativo el fichero de encabezamiento "msocket.h" que se utiliza en una implementación de ECTP.

```
#ifndef __MSOCKET_H_
#define __MSOCKET_H_
#include <sys/types.h>
```

```

#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/un.h>
#include "ectpcb.h"
#include "var.h"
#define SIZE_MSETSOCKOPT 524
struct _common{
    u_char command;
    u_short len;
    u_long error;
};
enum{ MCONNECT = 101,
    MBIND,
    MSEND,
    MRECV,
    MACCEPT,
    MLISTEN,
    MCLOSE,
    MSETSOCKOPT,
    MGETSOCKOPT,
    DETACH};

/* msocket return values */
#define ECONNECT_OK      0x00
#define ECONNECT_ROLE 0x01
#define ESEND_OK      0x0000
#define ESEND_RESEND 0x1004
#define ESEND_NOSEND 0x1001
#define ESEND_NOONE    0x1002
#define EBIND_OK      0x00
#define EBIND_ADDR 0x01
#define EBIND_ROLE 0x02

#define ECLOSE_OK      0x00
#define ECLOSE_TO      0x01 // When TO invoked
#define ECTP_QOS      0x4001
#define ECTP_OPVAR1 0x4002
#define ECTP_OPVAR2 0x4003
#define ECTP_OPLJ      0x4004
#define ECTP_OPWCR 0x4005
#define ECTP_OPTREE 0x4006
#define ECTP_OPTIME 0x4007
#define ECTP_OPFLOW      0x4008
#define ECTP_OPPAR 0x4009

/* msocket message types */

```

```

#define FIXED_SIZE    24    // size of fixed_header
struct _mbind_req{
    struct _common c;
    int    role;
    struct sockaddr_in ctrl;
    struct sockaddr_in local;
    struct sockaddr_in grp;

};
struct _mconnect_req{
    struct _common c;
};
struct _maccept_req{
    struct _common c;
    struct sockaddr_in to;
};
struct _msend_rep{
    struct _common c;
};
struct _mrecv_req{
    struct _common c;
};
struct _msend_req{
    struct _common c;
    u_char rsvd[FIXED_SIZE – sizeof(struct _common)];
    u_char  data[MAXDATA];
};
struct _mrecv_rep{
    struct _common c;
    u_char rsvd[FIXED_SIZE – sizeof(struct _common)];
    u_char  data[MAXDATA];
};
struct _mclose_req{
    struct _common c;
};
struct _mopt_op{
    int agn;
    int arn;
    int ccn;
    int ctl;
    int mcn;
    int mrn;
    int mtl;
    int nft;
    int lrtt;
};

```

```

struct _mopt_time{
    long agt; // ACK Generation time
    long cct; // Connection Creation time
    long hgt; // HB Generation time
    long iat; // Inactivity time
    long ndt; // ND Generation time
    long rbt; // Retransmission Backoff time
    long rxt; // Retransmission time
    long tct; // Tree Creation time
};

struct _mopt_flow{
    int cit;
    int cmn;
    int crn;
    int mintr;
    int ctr;
    int maxtr;
    int iri;
    int cri;
    int crd;
};

struct _mopt_tree{
    int tree_ct; // connection type
    int tree_conf; // Tree Level Configuration
    int tree_mtl; // Max Tree Level (only valid if tree_conf==2)
    int tree_mcn; // Max Children Number
    int bitmapsiz; // ack bitmap size
};

struct _mgetsockopt_req{
    struct _common c;
    int optname;
    int optlen;
    u_char optval[MAXDATA/2];
};

struct _msetsockopt_req{
    struct _common c;
    int optname;
    int optlen;
    u_char optval[MAXDATA/2];
};

char cbreq[128];
char dbreq[4096];
#endif

```

Bibliografía

Las siguientes RFC de IETF son útiles para comprender o implementar esta especificación:

- IETF RFC 768, *User Datagram Protocol, Internet Standard*, agosto de 1980.
- IETF RFC 791, *Internet Protocol*, DARPA Internet Program, Protocol specification, *Internet Standard*, septiembre de 1981.
- IETF RFC 793, *Transmission Control Protocol*, DARPA Internet Program, Protocol specification, *Internet Standard*, septiembre de 1981.
- IETF RFC 1112, *Host Extensions for IP Multicasting*, *Internet Standard*, agosto de 1989.
- IETF RFC 1119, *Network Time Protocol*, *Internet Standard*, mayo de 1990.
- IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, *Best Current Practice*, marzo de 1997.
- IETF RFC 2205, *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*, *Proposed Standard*, septiembre de 1997.
- IETF RFC 2210, *The Use of RSVP with IETF Integrated Services*, *Proposed Standard*, septiembre de 1997.
- IETF RFC 2236, *Internet Group Management Protocol, Version 2*, *Proposed Standard*, noviembre de 1997.
- IETF RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*, *Draft Standard*, diciembre de 1998.
- IETF RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, *Proposed Standard*, diciembre de 1998.
- IETF RFC 2597, *Assured Forwarding PHB Group*, *Proposed Standard*, junio de 1999.
- IETF RFC 2598, *An Expedited Forwarding PHB*, *Proposed Standard*, junio de 1999.
- IETF RFC 2750, *RSVP Extensions for Policy Control*, *Proposed Standard*, enero de 2000.
- IETF RFC 2836, *Per Hop Behavior Identification Codes*, *Proposed Standard*, mayo de 2000.

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	Redes de cable y transmisión de programas radiofónicos y televisivos, 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 Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación