Capítulo 5

Especificación del protocolo

Es necesario definir de manera precisa la operación del protocolo CBT a efectos de lograr una implementación correcta. Para realizar esta definición se toma como base la descripción en lenguaje natural provista en [7], y se la expresa en forma de máquina de estados finitos utilizando una notación basada en tablas evento/estado. Esta notación corresponde a la propuesta y utilizada por la ISO/OSI[30] para especificar la operación de las entidades que conforman un protocolo.

Los documentos de especificación de protocolos producidos por ISO/OSI consideran dos aspectos, la definición formal de la unidades de datos del protocolo (PDUs), y la especificación de su operación.

La definición de las unidades de datos del protocolo, o sintaxis del protocolo, se refiere al aspecto estático del mismo y se realiza en ASN.1 (Abstract Syntax Notation One), una notación estandarizada para descripción de datos que soporta sintaxis variadas.

La especificación de la operación de un protocolo describe las reglas que caracterizan el comportamiento del mismo. Este es el aspecto más complejo debido a que contempla interacciones entre diferentes entidades del mismo nivel o niveles adyacentes de la arquitectura.

En este caso particular, no es de importancia la definición de las PDUs debido a la simplicidad de su formato y su realización sólo sería utilizable como una manera no ambigua de especificación de las mismas[31]. Sí es de importancia la especificación de la operación del protocolo, la cual se realiza con el objetivo de disponer de una base que constituya una guía en la fase de implementación. Debe aclararse que la especificación realizada no tiene por objeto producir derivaciones automáticas de código ni posibilitar el análisis del protocolo respecto a su correctitud u otros aspectos a los que apuntan las técnicas formales de descripción (FDT).

5.1 Método utilizado para la especificación

La definición precisa de la operación del protocolo se realiza través de un máquina de estados finitos (FSM) que representa a la entidad CBT. Se describen sus estados, los eventos a los cuales reacciona y sus acciones, internas y externas (envío de PDUs). Para materializar esta representación se utiliza una tabla que contiene todas las combinaciones posibles de estados y eventos, indicando para cada una de ellas, posibles 3-uplas cuyas componentes representan un conjunto de acciones internas (IA) (por ejemplo cambio en el valor de una variable), un conjunto de eventos de salida (OE) (por ejemplo envío de una PDU) y el nuevo estado en que quedará el autómata (NS).

Los eventos, como se muestra en la figura 5.1, se agrupan en eventos provenientes del nivel superior (1), eventos provenientes de entidades pares remotas (2) y eventos internos provenientes de entidades locales (3).

[30] International Organization for Standardization, Open Systems Interconnection.

[31] El caso de el desarrollo de aplicaciones (Nivel Aplicación) en el contexto ISO/OSI es diferente, debido a la existencia de compiladores ASN.1 a diferentes lenguajes y al uso de una sintaxis común de transferencia (sintaxis concreta).

Dado un estado y un evento de entrada, podrán tomarse distintos conjuntos de acciones, según el estado de ciertas variables y la evaluación de ciertas condiciones. El conjunto

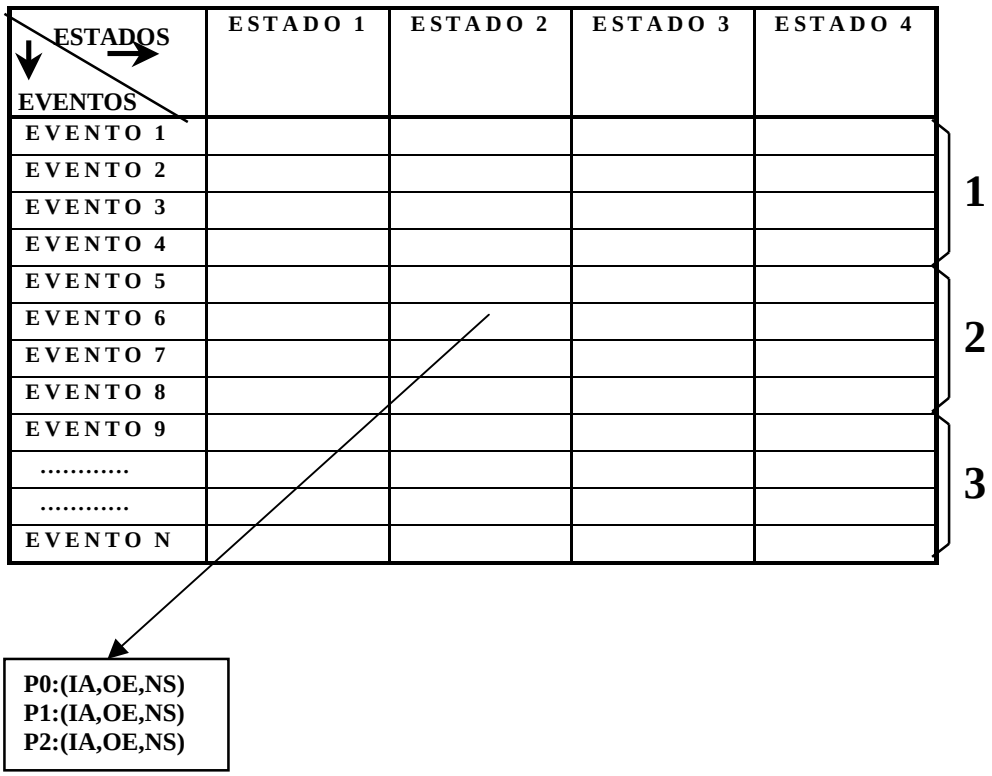

Fig 5.1. Formato de las tablas estado/evento. Utilizadas para describir la operación del protocolo

específico de acciones a tomar (definido por la 3-upla correspondiente), se basa en el valor de verdad de un predicado asociado al mismo. El conjunto de predicados de una entrada, debe cubrir todos los casos posibles.

5.2 Operación del protocolo

Previo a la descripción de los elementos de la FSM (estados, eventos, acciones internas y predicados), se describen los objetos internos con los cuales interactúa la entidad CBT: interfaces, timers y tablas.

5.2.1 Interfaces y vínculos

Una interfaz física está compuesta de un dispositivo de hardware (por ejemplo una placa de red) y el software asociado (driver del sistema operativo).

Una interfaz de este tipo está asociada con un vínculo físico, que puede ser por ejemplo un vínculo RS-232, una red Ethernet, etc. A continuación se describen las propiedades de interés para esta descripción que presentan dichas interfaces.

Desde el punto de vista de la cantidad de nodos que acceden al vínculo, tal como se lo percibe desde la interfaz física[32]:

- Interfaces a vínculos con capacidad multiacceso: desde la misma interfaz física, es posible acceder a diferentes equipos.
- Interfaces a vínculos sin capacidad multiacceso (punto a punto): desde la interfaz, sólo es posible acceder a un solo equipo.

Con relación al soporte de direcciones grupales[33]:

- Interfaces a vínculos con capacidad multicast: es posible utilizar direcciones grupales.
- Interfaces a vínculos sin capacidad multicast: sólo se permite el uso de direcciones de destino individuales.

Debe tenerse en cuenta que son válidas las cuatro combinaciones posibles:

- Interfaces a vínculos sin capacidad multiacceso y sin capacidad multicast.
- Interfaces a vínculos sin capacidad multiacceso con capacidad multicast
- Interfaces a vínculos con capacidad multiacceso sin capacidad multicast (redes NBMA).
- Interfaces a vínculos con capacidad multiacceso con capacidad multicast (por ejemplo redes Ethernet).

CBT puede operar sobre cualquier tipo de vínculo, y distingue el concepto de interfaz (CBT), diferente del concepto de interfaz física. Una interfaz CBT, se refiere a la interfaz física pero agrega una referencia al tipo de dirección (unicast o multicast) del router accedido a través de ella.

Una interfaz CBT se identifica, dentro del nodo, por la dirección IP de la interfaz local[34] (transmisión multicast), o bien por la dirección IP del nodo (upstream o downstream) (transmisión unicast). Debe considerarse que al identificar una interfaz CBT mediante una dirección de red, ya sea local o remota, se está haciendo referencia a una interfaz física local, la asociada a dicha dirección. De esta manera, pueden corresponder varias interfaces CBT a una interfaz física.

Sobre un vínculo punto a punto, se tendrá una interfaz CBT identificada por la dirección local si el vínculo soporta multicast, o una interfaz CBT identificada por la dirección del equipo remoto en caso contrario.

Sobre un vínculo multiacceso sin soporte multicast, se tendrá un conjunto de interfaces CBT cada una de ellas identificada por la dirección del equipo remoto.

Sobre un vínculo multiacceso con soporte multicast, se debe distinguir dos casos, existirá una interfaz (multicast) en la que el parent será el DR y los posibles child los demás nodos; esta interfaz se identificará en cada nodo, por su propia dirección. Puede existir una interfaz (unicast), en la cual el nodo DR es child y uno de los nodos no DR es parent; este caso se dará si el mejor camino al core del grupo no es el DR. Esta interfaz unicast, será identificada en el nodo DR a través de la dirección del "best hop" y en el nodo "best hop" a través de la dirección del nodo DR. De esta manera, el tráfico de PDUs unicast es ignorado por el resto de los nodos en el vínculo.

---

[32] Esta característica está dada por el tipo de subred asociada a la interfaz.

[33] Esta característica estará dada por las alternativas que ofrezca el protocolo de nivel de red (por ejemplo routers IP pueden o no soportar direccionamiento multicast) o por las facilidades ofrecidas por la subred.

[34] En el presente trabajo no se considera la utilización de varias redes lógicas -direcciones de redes IP- sobre un vínculo.

Para evitar ciclos en el árbol de distribución, un nodo debe comprobar al agregar una interfaz child que ella no coincida con la parent del grupo. Una comprobación adicional que debe realizar un nodo no DR es que la red asociada a cualquiera de sus interfaces child para el grupo sea diferente de la asociada a su parent para evitar que un nodo al cual el DR elige como parent produzca un ciclo en el caso de seleccionar el vínculo multiacceso como mejor camino al core. Un nodo actuando como DR, en cambio, puede tener una interfaz parent (unicast) sobre la misma red que algunas de sus child (multicast), en el caso de que el mejor camino al core esté constituido por otro router del vínculo multiacceso.

Una interfaz CBT almacena, además de la dirección y tipo de envío (unicast o multicast), otro tipo de información, tal como estado de DR o no, valor de preferencia (no

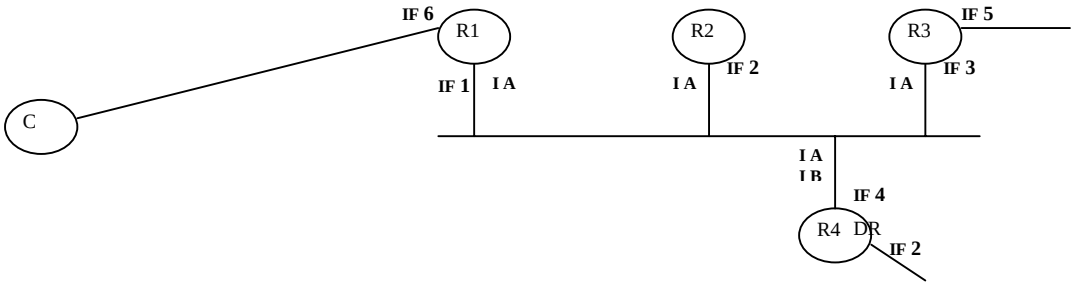

Figura 5.2. Relación entre interfaces físicas e interfaces CBT.

utilizado en la especificación debido a que no se implementa el protocolo HELLO), etc.

En la figura 5.2 se muestra una red multiacceso con capacidad multicast a la cual están conectados 4 routers, R1, R2, R3 y R4. R4 actúa como DR mientras que R1 constituye el mejor nodo desde el punto de vista del vínculo multiacceso para enviar al core de un grupo determinado, C. Las interfaces físicas se indican como IF x, siendo esta identificación la dirección de red local en el vínculo, mientras que las interfaces CBT se indican como I x.

Se muestran a continuación las interfaces CBT en cada uno de los nodos y cómo son identificadas a partir de las correspondientes interfaces físicas (o direcciones de las placas de red):

Router R4:
- IA: interfaz child sobre el vínculo multiacceso. IF4,M
- IB: interfaz parent al grupo (a R1): IF4, IF1

Router R3:
- IA:interfaz parent sobre el vínculo multiacceso. IF3,M

Router R2:
- IA:interfaz parent sobre el vínculo multiacceso. IF2,M

Router R1:
- IA:interfaz child sobre el vínculo multiacceso. IF1,IF4

El carácter unicast o multicast de una interfaz CBT está indicado por el segundo componente de su identificación, M (multicast) o la dirección del nodo adyacente (IFx).

Un error que produciría ciclos en el árbol de distribución sería que R1 constituyera una interfaz child sobre el vínculo multiacceso, de la misma manera que R2 o R3.

En adelante se hará referencia a una interfaz CBT utilizando el término "interfaz", y a una interfaz física utilizando el término "vínculo".

5.2.2 Tablas

Las tablas sobre las cuales opera la entidad CBT, representan la información multicast correspondiente al árbol de distribución (MFC: Multicast Forwarding Cache), que será utilizada para el mantenimiento del árbol y la distribución de los paquetes multicast. La información que contiene cada entrada, como fue visto en capítulos anteriores, consiste del grupo multicast, que constituye el índice de la tabla, la interfaz (CBT) parent (en dirección a la raíz del árbol) y un conjunto de interfaces child, hacia las hojas del árbol.

Existen dos tablas con información similar: la tabla de entradas transitorias, que contiene información respecto de grupos para los cuales se ha solicitado la pertenencia al árbol de distribución pero aún no ha sido confirmada, y la tabla permanente, que contiene aquellas entradas que indican interfaces que ya pertenecen al árbol de distribución (estado on-tree del nodo respecto del grupo).

Una información adicional contenida en cada interfaz child de cada entrada, se refiere a la condición de hoja del árbol. Dicha condición influye en la operación del protocolo.

La entidad CBT creará y eliminará entradas en las tablas de acuerdo a requerimientos recibidos, sus correspondientes respuestas y eventos tales como timeouts.

5.2.3 Timers

Los timers son utilizados por la entidad CBT para realizar reintentos ante operaciones no exitosas (p.ej. retransmisión de JOIN_REQUESTs), para la eliminación de entradas en las tablas transitorias y permanentes y para la repetición de procedimientos que deben efectuarse periódicamente, tales como el monitoreo de las interfaces parent por parte de un nodo.

Las operaciones realizadas sobre los timers tenidas en cuenta en la especificación son las siguientes:

- **Activación**: El timer es cargado con un valor y comienza a decrementarlo. Cuando llegue a cero producirá el evento asociado con su expiración (timeout).
- **Desactivación**: El conteo del timer es interrumpido, evitando el timeout.
- **Reactivación**: El timer es desactivado y vuelto a activar con un nuevo valor. Se utiliza, por defecto el valor original.

Las operaciones anteriores son las que tienen relevancia en la especificación del protocolo. Es importante destacar que desde el punto de vista de una implementación, existen dos operaciones adicionales, la creación y la eliminación de un timer, que se refieren a la asignación y liberación de los recursos insumidos por el objeto. Es importante la liberación de recursos, ya que en este caso la cantidad de timers (y por lo tanto los recursos que se consumen) varía de acuerdo a la actividad en la red; por ejemplo, un nodo que esté incorporado al árbol de distribución de un cierto número de grupos, deberá tener una entrada permanente por cada uno de ellos, junto con su timer asociado; al dejar de pertenecer al árbol, tanto la entrada como el timer serán eliminados.

Otro aspecto a tener en cuenta, es la utilización de timers comunes a más de un grupo; es el caso del timer para producir las PDUs ECHO_REQUEST. Bastará con uno por interfaz

CBT, pudiendo éste servir a todos los grupos que compartan en el nodo dicha interfaz parent. En la descripción, por razones de simplicidad, se considerará un timer por grupo multicast.

A continuación se describen los timers utilizados en la especificación, indicando su tiempo asociado, tal como se especifica en [7], una breve descripción de la operatroria asociada con cada uno de ellos, su objetivo y las causas que producen que la entidad CBT invoque sus métodos de activación, reactivación o desactivación.

T01: Retransmit_Join_Request_Timer (RTX_INTERVAL): Una entidad CBT que crea una entrada transitoria leaf como consecuencia de recibir un requerimiento de agregarse a un grupo por parte de un proceso local o de una entidad IGMP tiene la responsabilidad de generar reintentos de conexión al grupo a través de la retransmisión de JOIN_REQUESTs en forma periódica.

- **Objetivo**: Provocar la retransmisión del JOIN_REQUEST si no se obtiene respuesta (JOIN_ACK) en un determinado tiempo, RTX_INTERVAL.
- **Activación**: Al crear la entrada transitoria y luego cada vez que se reenvía el JOIN_REQUEST.
- **Reactivación**: No.
- **Desactivación**: Cuando se elimina la entrada transitoria, ya sea porque se recibe respuesta (JOIN_ACK) o transcurre el tiempo máximo prefijado.

T02: Delete_Leaf_Transient_Entry_Timer (JOIN_TIMEOUT): Una entidad CBT que ha creado una entrada leaf transitoria establece un tiempo máximo de espera por la respuesta del parent (JOIN_ACK). Transcurrido este tiempo, la entrada es eliminada.

- **Objetivo**: Provocar la eliminación de la entrada leaf transitoria si no se recibe respuesta (JOIN_ACK) en el tiempo preestablecido (JOIN_TIMEOUT).
- **Activación**: Al crear la entrada transitoria.
- **Reactivación**: No.
- **Desactivación**: Cuando se elimina la entrada transitoria, ya sea porque se recibe respuesta (JOIN_ACK) o transcurre el tiempo máximo prefijado.

T03: Delete_Transient_Entry_Timer (TRANSIENT_TIMEOUT): Una entidad CBT que ha creado una entrada transitoria (no leaf) establece un tiempo máximo de espera por la respuesta del parent (JOIN_ACK). Transcurrido este tiempo, la entrada es eliminada.

- **Objetivo**: Provocar la eliminación de la entrada transitoria si no se recibe respuesta (JOIN_ACK) en el tiempo preestablecido (TRANSIENT_TIMEOUT).
- **Activación**: Al crear la entrada transitoria
- **Reactivación**: No.
- **Desactivación**: Cuando se elimina la entrada transitoria, ya sea porque se recibe respuesta (JOIN_ACK) o transcurre el tiempo máximo prefijado.

T04: Cache_Deletion_Timer (CACHE_DEL_TIMER): Una entidad CBT que recibe un QUIT_NOTIFICATION por una interfaz child multicast, no la elimina inmediatamente, ya que pueden existir otros nodos child que deseen permanecer en el grupo. Para dar tiempo a que estos nodos lo soliciten (a través de un JOIN_REQUEST), espera un tiempo predeterminado, dado por este timer.

- **Objetivo**: Provocar la eliminación de la interfaz child multicast asociada si luego de un período de tiempo (CACHE_DEL_TIMER) no es recibido un JOIN_REQUEST de algún nodo child.
- **Activación**: Al recibir el primer QUIT_NOTIFICATION por una interfaz child multicast.
- **Reactivación**: No.

- **Desactivación**: Cuando se elimina la entrada child o cuando se recibe un JOIN_REQUEST por la interfaz asociada.

T05: Transmit_Join_Request_Timer (Random [0, HOLDTIME]): Una entidad CBT que recibe un QUIT_NOTIFICATION por una interfaz parent multicast y desea seguir perteneciendo al grupo, debe enviar un JOIN_REQUEST al parent para que éste no elimine la interfaz.

- **Objetivo**: Provocar el envío de un JOIN_REQUEST por la interfaz correspondiente, en un tiempo generado al azar entre 0 y HOLDTIME.
- **Activación**: Al recibir un QUIT_NOTIFICATION por la interfaz parent multicast.
- **Reactivación**: No.
- **Desactivación:** Cuando se cumple el tiempo y se envía el JOIN_REQUEST o cuando se recibe un JOIN_REQUEST (de otro nodo child) por la interfaz parent multicast.

T06: Quit_Notification_Timer (HOLDTIME): Una entidad que desea dejar de integrar el grupo, envía un QUIT_NOTIFICATION al parent. Debido a que esta PDU no tiene asentimiento, es enviada un cierto número de veces a intervalos regulares.

- **Objetivo**: Provocar la retransmisión del QUIT_NOTIFICATION a intervalos regulares (HOLDTIME).
- **Activación**: Al eliminar una entrada permanente y enviar el primer QUIT_NOTIFICATION al parent y luego al realizar cada una de las retransmisiones.
- **Reactivación**: No.
- **Desactivación**: Al concluir la cuenta de reenvíos.

T07: Echo_Request_Timer (ECHO_INTERVAL): Una entidad CBT debe comprobar periódicamente el funcionamiento normal de su parent (para cada grupo). Lo hace enviando ECHO_REQUEST a intervalos regulares (ECHO_INTERVAL); a cada ECHO REQUEST debe responder el parent con un ECHO_REPLY.

- **Objetivo**: Producir el envío de un ECHO_REQUEST.
- **Activación**: Al crear una entrada permanente y luego al enviar cada ECHO_REQUEST.
- **Reactivación**: Cuando se recibe un ECHO_REQUEST por la interfaz parent (caso multicast); de esta manera se evita saturar el vínculo con envíos innecesarios.
- **Desactivación**: Al eliminar la entrada permanente.

T08: Delete_On-tree_Entry_Timer (GROUP_EXPIRE_TIME): Una entidad CBT que ha creado una entrada permanente (on-tree) establece un tiempo máximo de espera para ECHO_REPLYs enviados por el parent (GROUP_EXPIRE_TIME). Transcurrido este tiempo, si no se ha recibido el ECHO_REPLY, la entrada es eliminada.

- **Objetivo**: Provocar la eliminación de la entrada permanente si no se registra actividad del parent en el tiempo preestablecido.
- **Activación**: Al crear la entrada permanente.
- **Reactivación**: Al recibir un ECHO_REPLY por la interfaz parent.
- **Desactivación**: Cuando se elimina la entrada permanente.

T09: Echo_Reply_Timer (random[0,HOLDTIME]): Una entidad CBT que recibe un ECHO_REQUEST por una interfaz child, debe responder en un plazo máximo (HOLDTIME) con un ECHO_REPLY para indicar a su(s) child que está activa.

- **Objetivo**: Provocar el envío de un ECHO_REPLY en una interfaz child en la que se recibió un ECHO_REQUEST, en un tiempo al azar, entre 0 y HOLDTIME después de haber recibido el ECHO_REQUEST.

- **Activación**: Al recibir un ECHO_REQUEST por una interfaz child (si está inactivo).
- **Reactivación**: No.
- **Desactivación**: Cuando se elimina la entrada child.

5.2.4 Descripción de los elementos de la FSM

A continuación se describen los estados, eventos de entrada, condiciones que conforman los predicados, acciones internas y eventos externos generados por la entidad CBT. La descripción se realiza en base a un grupo multicast.

5.2.4.1 Estados

Se considera que la entidad CBT puede estar en los siguientes estados respecto a un grupo multicast:

ES01: Estado on-tree: El nodo pertenece al árbol de distribución CBT. La entidad CBT debe responder a requerimientos de integrar el árbol provenientes de nodos adyacentes y mantener la conectividad con su parent y sus child.

ES02: Estado transitorio: La entidad CBT ha recibido un requerimiento, ya sea de una entidad local (proceso de aplicación), de una entidad IGMP o de una entidad CBT adyacente para integrar el árbol de distribución de un grupo, y como consecuencia ha iniciado y aún no completado el procedimiento de integración a un árbol de distribución. Es responsable de completar este procedimiento o abortarlo después de un cierto tiempo.

ES03: Estado sin entrada: El nodo no pertenece al árbol de distribución del grupo ni ha iniciado procedimientos para integrarlo.

5.2.4.2 Eventos de entrada

Los eventos de entrada se clasifican de la siguiente manera, según su origen:

- Eventos externos, provenientes de entidades CBT remotas, que interatcúan con la entidad local a través de PDUs CBT.
- Eventos provenientes de otras entidades en el nodo: aplicaciones en el nodo que se unen o abandonan los grupos multicast y entidades IGMP que solicitan unirse o abandonar grupos en representación de hosts de subredes locales; la interacción se produce a través de los métodos join-group y leave-group.
- Eventos internos del protocolo, producidos por los timers asociados al mismo.

5.2.4.2.1 Eventos externos provenientes de entidades CBT remotas

Este tipo de eventos ocurre como consecuencia del arribo de PDUs. Su efecto sobre la entidad que los recibe depende de condiciones tales como el estado de la entidad receptora, tipo de vínculo por el que se los recibe (capacidad multicast o no) y tipo de envío (unicast o multicast).

EV01: Recepción de JOIN_REQUEST: La ocurrencia de este evento puede tener dos significados, según se detalla a continuación:

- El nodo receptor recibe la PDU como consecuencia de que otro nodo que aún no integra el árbol de distribución del grupo desea integrarlo, siendo el receptor el próximo nodo en el camino al core (según el emisor). El nodo que lo recibe, podrá pertenecer al árbol de distribución del grupo, caso en el que responderá con un JOIN_ACK, o no, caso en el que repetirá el procedimiento realizado por el nodo anterior.
- El nodo recibe la PDU sobre una interfaz child multiacceso para el grupo, como consecuencia de que otro nodo ha detectado que un tercero desea podar el árbol, mientras que aquél desea mantenerlo.

EV02: Recepción de JOIN_ACK: Es recibido en respuesta a un JOIN_REQUEST enviado previamente. Debe ser recibido por la interfaz (CBT) por la cual se ha enviado el JOIN-REQUEST correspondiente. Pone al nodo en estado on-tree, haciendo que éste integre el árbol de distribución para el grupo.

EV03: Recepción de QUIT_NOTIFICATION: Es recibido como consecuencia de que un nodo child notifica al nodo (su parent) que desea dejar de pertenecer al árbol de distribución de un grupo.

EV04: Recepción de FLUSH_TREE: Se recibe como consecuencia de que el nodo parent desea eliminar el subárbol de distribución que de él depende. Puede ser consecuencia de que ha perdido contacto con su parent o ha recibido a su vez un FLUSH_TREE de éste.

EV05: Recepción de ECHO_REQUEST: Se recibe como consecuencia de la comprobación periódica llevada a cabo por un nodo child para confirmar la conectividad con el parent.

EV06: Recepción de ECHO_REPLY: Es la respuesta a un ECHO_REQUEST enviado previamente; el nodo parent confirma que está operativo.

5.2.4.2.2 Eventos producidos por otras entidades residentes en el nodo

Este tipo de eventos se produce como consecuencia de requerimientos realizados por dos tipos de entidades: procesos de aplicación y entidades IGMP, ambas residentes en el nodo.

Un proceso de aplicación realiza la solicitud de integrar un grupo multicast en virtud de su funcionalidad, mientras que una entidad local IGMP lo hace en función de haber recibido un requerimiento de un host en la subred local.

A efectos de ignorar detalles de implementación, se considera que la indicación a la entidad CBT se produce a través de los eventos genéricos join-group y leave-group.[35], es decir, no se hace diferencia en cuanto a los mecanismos para la interacción entre entidades representando a aplicaciones locales y entidades IGMP con el agente CBT, utilizando las primitivas join-group y leave-group para solicitar integrar o dejar de pertenecer a un grupo respectivamente.

[35] En una implementación concreta, si este evento proviene de IGMP, éste lo anunciará al protocolo de ruteo a través de las acciones "notify routing +" y "notify routing –" definidas en IGMPv2[2]. En el medioambinte de simulación, pese a que no existen aún implementaciones de IGMP, es razonable que tomen la forma de agentes y que invoquen a métodos CBT que se correspondan con los eventos join-group y leave-group, de igual manera que se implementa para las aplicaciones multicast en el nodo.

EV11: Recepción de JOIN_GROUP: Se recibe como consecuencia de que una entidad residente en el nodo (IGMP o proceso de aplicación) solicita integrar un grupo multicast. Como posible consecuencia, si el nodo no integra el árbol de distribución del grupo, se iniciará el procedimiento correspondiente, y en este caso se creará una entrada transitoria leaf.

EV12: Recepción de LEAVE-GROUP: Se recibe como consecuencia de que una entidad residente en el nodo (IGMP o proceso de aplicación) desea dejar de pertenecer a un grupo multicast. Como posible consecuencia, la entidad CBT podrá requerir que se pode el árbol de distribución del grupo.

5.2.4.2.3 Eventos locales

Estos eventos se producen como consecuencia de la expiración (timeout) de los diferentes timers controlados por la entidad CBT. Para una explicación más detallada, referirse a la sección timers.

EV21: Vencimiento de Retransmit_Join_Request_Timer (T01)

EV22: Vencimiento de Delete_Leaf_Transient_Entry_Timer (T02)

EV23: Vencimiento de Delete_Transient_Entry_Timer (T03)

EV24: Vencimiento de Cache_Deletion_Timer (T04)

EV25: Vencimiento de Transmit_Join_Request_Timer (T05)

EV26: Quit_Notification_Timer (T06)

EV27: Vencimiento de Echo_Request_Timer (T07)

EV28: Vencimiento de Delete_On-tree_Entry_Timer (T08)

EV24: Vencimiento de Echo_Reply_Timer (T09)

5.2.4.3 Acciones internas

Las acciones internas se seleccionan de manera tal que describan la operación con una granularidad que permita una descripción simple y no ambigua.

Para el caso de CBT, las acciones internas afectan a las tablas que representan el árbol de distribución, (ya sea información transitoria o permanente), el estado de los timers (activo, inactivo) y los valores de variables internas del protocolo (por ejemplo, contadores de reintentos).

Se trata, a nivel de la implementación, que los métodos de las clases que componen el protocolo CBT en el medioambiente de la simulación estén estrechamente relacionadas con estas acciones.

Las acciones internas, son las siguientes:

IA000: Crear entrada transitoria para un grupo.
IA001: Eliminar entrada transitoria para un grupo.
IA002: Agregar child en entrada transitoria.
IA003: Eliminar child en entrada transitoria.

IA010: Crear entrada permanente (on-tree) para un grupo.
IA011: Eliminar entrada permanente (on-tree) para un grupo.
IA012: Agregar child en entrada permanente (on-tree).
IA013: Eliminar child en entrada permanente (on-tree).

IA020: Activar Retransmit_Join_Request_Timer (T01)
IA021: Desactivar Retransmit_Join_Request_Timer (T01)
IA022: Reactivar Retransmit_Join_Request_Timer (T01)

IA030: Activar Delete_Leaf_Transient_Entry_Timer (T02)
IA031: Desactivar Delete_Leaf_Transient_Entry_Timer (T02)
IA032: Reactivar Delete_Leaf_Transient_Entry_Timer (T02)

IA040: Activar Delete_Transient_Entry_Timer (T03)
IA041: Desactivar Delete_Transient_Entry_Timer (T03)
IA042: Reactivar Delete_Transient_Entry_Timer (T03)

IA050: Activar Cache_Deletion_Timer (T04).
IA051: Desactivar Cache_Deletion_Timer (T04).
IA052: Reactivar Cache_Deletion_Timer (T04).

IA060: Activar Transmit_Join_Request_Timer (T05).
IA061: Desactivar Transmit_Join_Request_Timer (T05)
IA062: Reactivar Transmit_Join_Request_Timer (T05)

IA070: Activar Quit_Notification_Timer (T06)
IA071: Desactivar Quit_Notification_Timer (T06).
IA072: Reactivar Quit_Notification_Timer (T06).

IA080: Activar Echo_Request_Timer (T07).
IA081: Desactivar Echo_Request_Timer (T07).
IA082: Reactivar Echo_Request_Timer (T07)

IA090: Activar Delete_On-tree_Entry_Timer (T08)
IA091: Desactivar Delete_On-tree_Timer (T08).
IA092: Reactivar Delete_On-tree_Timer (T08).

IA100: Activar Echo_Reply_Timer (T09).
IA101: Desactivar Echo_Reply_Timer (T09).
IA102: Reactivar Echo_Reply_Timer (T09).

IA200: Variable RTX = RTX + 1
IA201: Variable RTX = 0

5.2.4.4 Eventos de salida (OE)

Los eventos de salida consisten en el envío de PDUs CBT a entidades pares CBT residentes en otros nodos.

Los eventos de salida, son los siguientes:

OE01: Enviar JOIN_ACK
OE02: Enviar JOIN_REQUEST

OE03: Enviar QUIT_NOTIFICATION
OE04: Enviar FLUSH_TREE
OE05: Enviar ECHO_REQUEST
OE06: Enviar ECHO_REPLY

5.2.4.5 Predicados

Un predicado representa la evaluación de ciertas condiciones acerca del estado de variables internas y otras tales como el tipo de interfaz a través de la cual arriba una PDU.

Según sea el valor del predicado verdadero o falso, se ejecuta o no el conjunto de acciones (3-upla) asociado a él.

El conjunto de predicados especificado para un par (estado, evento) debe cumplir con las condiciones siguientes:

- Sólo uno de los predicados puede tener valor verdadero.
- El conjunto de los predicados debe cubrir todos los posibles casos.

En la presente especificación se asume que si ninguno de los predicados que figuran explícitamente tiene valor verdadero, el evento es igonrado, es decir IA = OE = ∅ (conjunto vacío) y NS = estado actual.

El objeto de los predicados es evitar la explosión de estados del autómata que tendría lugar si no se considerara el uso de variables internas. Las condiciones que componen los predicados, se han seleccionado de manera que la especificación resulte clara y no dependa de detalles de implementación.

5.2.4.6 Condiciones

Las condiciones que conforman un predicado se refieren a sentencias con valores verdadero o falso que representan por ejemplo estados de contadores, tipo de un vínculo, estados de una interfaz o del nodo (DR, no DR), etc. En una implementación, no es necesaria la existencia de variables específicas que representen dichos valores.

C01: Interfaz de arribo de la PDU es parent (transitoria o permanente) para el grupo.
C02: Interfaz de arribo de la PDU es child (transitoria o permanente) para el grupo.
C03: Entrada transitoria es leaf.
C04: Vínculo de arribo de la PDU es multiacceso.
C05: Vínculo de arribo de la PDU es multicast.
C06: Interfaz de arribo es la única child para el grupo.
C07: Existe interfaz correcta a nodo upstream en dirección al core del grupo.
C08: Vínculo asociado a la interfaz parent del grupo (permanente o transitoria) es multiacceso.
C09: Vínculo asociado a la interfaz parent del grupo (permanente o transitoria) es multicast.
C10: Nodo es DR en el vínculo asociado a la interfaz.
C11: Red asociada a la interfaz parent es diferente de red asociada a la interfaz de arribo.
C12: Interfaz a nodo upstream es diferente de interfaz de arribo.
C13: La PDU recibida lleva dirección multicast.
C14: Contador de reintentos no llegó al límite (RTX < MAX_RTX).

	ON TREE (ES01)	TRANSITORIO (ES02)	SIN ENTRADA (ES03)
(EV01) Recepción JOIN_REQUEST	P01 P06 P02 P07 P03 P08 P04 P09 P05 P10	P11 P12 P13 P14 P15	P16 P17 P18 P19 P20
(EV02) Recepción JOIN_ACK	IGNORAR	P21 P22	IGNORAR
(EV03) Recepción QUIT_NOTIFICATION	P23 P24 P25 P26	IGNORAR	IGNORAR
(EV04) Recepción FLUSH_TREE	P27 P28	IGNORAR	IGNORAR
(EV05) Recepción ECHO_REQUEST	P29 P30	IGNORAR	IGNORAR
(EV06) Recepción ECHO_REPLY	P31	IGNORAR	IGNORAR
(EV10) Recepción JOIN-GROUP	P32	P33	P34
(EV11) Recepción LEAVE-GROUP	P35 P36 P37	IGNORAR	IGNORAR

Tabla 5.1. Tabla de estados/eventos para eventos externos *(provenientes de otros nodos o de entidades en el nodo) a la entidad CBT*

	ON TREE (ES01)	TRANSITORIO (ES02)	SIN ENTRADA (ES03)
(EV21) Retransmit_Join_Request_Timer (T01) timeout	IGNORAR	P38	IGNORAR
(EV22) Delete_Leaf_Transient_Entry_Timer (T02) timeout	IGNORAR	P39	IGNORAR
(EV23) Delete_Transient_Entry_Timer (T03) timeout	IGNORAR	P40	IGNORAR
(EV24) Cache_Deletion_Timer (T04) timeout	P41 P42	IGNORAR	IGNORAR
(EV25) Transmit_Join_Request_Timer (T05) timeout	P43	IGNORAR	IGNORAR
(EV26) Quit_Notification_Timer (T06) timeout	IGNORAR	IGNORAR	P44 P45
(EV27) Echo_Request_Timer (T07) timeout	P46	IGNORAR	IGNORAR
(EV28) Delete_Ontree_Timer (T08) timeout	P47 P48 P49	IGNORAR	IGNORAR
(EV29) Echo_Reply_Timer (T09) timeout	P50	IGNORAR	IGNORAR

Tabla 5.2. Tabla de estados/eventos para eventos internos *(provenientes de timers) de la entidad CBT*

5.2.4.7 Tablas evento/estado

Las tablas 5.1 y 5.2 muestran los posibles estados de la entidad CBT y los posibles eventos de entrada. De acuerdo con lo descripto al principio del capítulo, en cada caso se indica una referencia a los posibles predicados (compuestos por operaciones lógicas entre las condiciones ya enunciadas) y las acciones internas, externas y nuevo estado de la entidad.

La notación utilizada está compuesta de una referencia al casillero de la tabla, operación lógica entre los valores verdadero o falso de determinadas condiciones, seguida de la 3-upla que contiene referencias a acciones internas, eventos externos y nuevo estado. En caso de realizarse alguna acción para la totalidad de las entradas child, se lo indica con *C: seguido por el conjunto de acciones internas a realizar o eventos a producir.

Por ejemplo:

P50: C20 & C21 {*C:(IA050, IA051), ,(ES02)}

Indica referencia a P50; su predicado está dado por el valor de C20 y C21, y en caso de que esta operación produzca un resultado verdadero, se realiza, para cada una de las interfaces child del grupo, las acciones IA050 y IA051; no se producen eventos externos, y el nuevo estado es ES02. En el caso de una 3-upla que sea válida para cualquier conjunto de valores, se indica su predicado con el valor verdadero, V.

A continuación se indican las condiciones a evaluar, acciones internas y eventos de salida para cada una de las referencias de las tablas 5.1 y 5.2, junto con una explicación breve relativa al evento producido.

Recepción de JOIN_REQUEST en estado on-tree

P01: !C01 & !C02 & !C04 {(IA012), (OE01), (ES01)}
La PDU es recibida por una interfaz que aún no es parent ni child para el grupo (!C01 & !C02); el vínculo asociado no tiene capacidad multiacceso (!C04). En este caso se agrega la interfaz de arribo como child para el grupo (IA012). El evento a generar es un JOIN_ACK (OE01) direccionado según la interfaz de arribo. No hay cambio de estado.

P02: !C01 & !C02 & C04 & !C05 & C10 {(IA012), (OE01), (ES01)}
La PDU es recibida por una interfaz que aún no es parent ni child para el grupo (!C01 & !C02); el vínculo asociado tiene capacidad multiacceso (C04) no multicast (!C05) y el nodo es DR en el vínculo (C10). En este caso se agrega la interfaz de arribo como child para el grupo (IA012). El evento a generar consiste en la emisión de un JOIN_ACK direccionado según la interfaz de arribo (OE01). No se produce cambio de estado.

P03: !C01 & !C02 & C04 & !C05 & !C10 & C11 {(IA012), (OE01), (ES01)}
La PDU es recibida por una interfaz que aún no es parent ni child para el grupo (!C01 & !C02); el vínculo asociado es multiacceso (C04) no multicast (!C05) , el nodo no es DR en el vínculo (!C10), y la red a la que pertenece el nodo parent es diferente a la red de arribo (C11) (esta comprobación es necesaria para evitar ciclos en el árbol de distribución). En este caso se agrega la interfaz de arribo como child para el grupo (IA012). El evento externo consiste en el envío de un JOIN_ACK direccionado según la interfaz de arribo (OE01). No se produce cambio de estado.

P04: !C01 & !C02 & C04 & C05 & C10 & C13 {(IA102), (OE01), (ES01)}
La PDU es recibida por una interfaz que aún no es parent ni child para el grupo (!C01 & !C02); el vínculo asociado tiene capacidad multiacceso (C04) multicast (C05); y el nodo es DR en el vínculo (C10). La PDU recibida lleva dirección multicast (C13). En este caso se agrega la interfaz de arribo como child para el grupo (IA012). El evento a generar es el envío de un JOIN_ACK direccionado según la interfaz de arribo (OE01). No se produce cambio de estado.

P05: !C01 & !C02 & C04 & C05 & !C10 & !C13 & C11 {(IA102), (OE01), (ES01)}
La PDU es recibida por una interfaz que aún no es parent ni child para el grupo (!C01 & !C02); el vínculo asociado tiene capacidad multiacceso (C04) multicast (C05); y el nodo no es DR en el vínculo (!C10). La PDU recibida lleva dirección unicast (!C13). La red a la que pertenece el nodo parent es diferente a la red de arribo (C11). En este caso se agrega la interfaz de arribo como child para el grupo (IA012). El evento a generar es el envío de un JOIN_ACK direccionado según la interfaz de arribo (OE01). No se produce cambio de estado.

P06: C01 & C04 & C05 & !C10 {(IA061), , (ES01)}
La PDU es recibida por la interfaz parent para el grupo (C01), el vínculo asociado es multiacceso (C04) multicast (C05) y el nodo no es DR en el vínculo (!C10). En este caso, se debe desactivar, si está activo, el timer para envío de JOIN_REQUEST (IA061) (T05). No se generan eventos externos. No hay cambio de estado.

P07: C02 & !C04 { , (OE01), (ES01)}
La PDU es recibida por una interfaz child para el grupo (C02); el vínculo asociado no posee característica multiacceso (!C04). En este caso no se producen acciones internas. Se genera un evento externo consistente en el envío de un JOIN_ACK direccionado según la interfaz de arribo (OE01). No hay cambio de estado.

P08: C02 & C04 & !C05 { , (OE01), (ES01)}
La PDU es recibida por una interfaz child para el grupo (C02); el vínculo asociado es multiacceso (C04) no multicast (!C05). En este caso no se realizan acciones internas. Se genera un evento externo consistente en el envío de un JOIN_ACK según la interfaz de arribo (OE01). No se cambia de estado.

P09: C02 & C04 & C05 & C10 {(IA051), (OE01), (ES01)}
La PDU es recibida por una interfaz child para el grupo (C02); el vínculo asociado es multiacceso (C04) multicast (C05) y el nodo es DR en el vínculo (C10). En este caso se desactiva el timer para borrado de la interfaz child (IA051) (T04) si es que estaba activo. Como evento externo, se envía un JOIN_ACK direccionado según la interfaz de arribo (OE01). No se produce cambio de estado.

P10: C02 & C04 & C05 & !C10 { , (OE01), (ES01)}
La PDU es recibida por una interfaz child para el grupo (C02); el vínculo asociado es multiacceso (C04) multicast (C05) y el nodo no es DR en el vínculo (!C10). En este caso no se realizan acciones internas. Como evento externo, se envía un JOIN_ACK direccionado según la interfaz de arribo (OE01). No se produce cambio de estado.

Recepción de JOIN_REQUEST en estado transitorio

P11: !C01 & !C02 & !C04 {(IA002), , (ES02)}
La PDU es recibida por una interfaz que aún no es parent (!C01) ni child (!C02) para el grupo y el vínculo asociado no tiene capacidad multiacceso (!C04). En este caso se agrega la interfaz de arribo como child para el grupo transitorio (IA002). No se producen eventos externos. No se cambia de estado.

P12: !C01 & !C02 & C04 & !C05 & C10 {(IA002), , (ES02)}
La PDU es recibida por una interfaz que aún no es parent (!C01) ni child (!C02) para el grupo, el vínculo asociado es multiacceso (C04) no multicast (!C05) y el nodo es DR en el vínculo (C10). En este caso se agrega la interfaz de arribo como child para el grupo transitorio (IA002). No se producen eventos externos. No se cambia de estado.

P13: !C01 & !C02 & C04 & !C05 & !C10 & C11 {(IA002), ,(ES02)}
La PDU es recibida por una interfaz que aún no es parent (!C01) ni child (!C02) para el grupo, el vínculo asociado es multiacceso (C04) no multicast (!C05), el nodo no es DR en el vínculo (!C10) y la red a la que pertenece el nodo parent es diferente a la red de arribo de la PDU (C11). En este caso se agrega la interfaz de arribo como child para el grupo transitorio (IA002). No se producen eventos externos. No se cambia de estado.

P14: !C01 & !C02 & C04 & C05 & C10 & C13 {(IA002), , (ES02)}
La PDU es recibida por una interfaz que aún no es parent (!C01) ni child (!C02) para el grupo, el vínculo asociado es multiacceso (C04) multicast (C05); el nodo es DR en dicho vínculo (C10) y la PDU ha sido enviada con dirección multicast (C13). En este caso se agrega la interfaz de arribo como child para el grupo transitorio (IA002). No se producen eventos externos. No se cambia de estado.

P15: !C01 & !C02 & C04 & C05 & !C10 & !C13 & C11 {(IA002), , (ES02)}
La PDU es recibida por una interfaz que aún no es parent (!C01) ni child (!C02) para el grupo, el vínculo asociado es multiacceso (C04) multicast (C05); el nodo no es DR en dicho vínculo (!C10) y la PDU ha sido enviada con dirección unicast (!C13). La red asociada a la interfaz parent es diferente de la red asociada a la interfaz de arribo de la PDU (C11). En este caso se agrega la interfaz de arribo como child para el grupo transitorio (IA002). No se producen eventos externos. No se cambia de estado.

Recepción de JOIN_REQUEST en estado sin entrada

P16: !C04 & C07 & C12 {(IA000, IA040, IA002, IA071), (OE02), (ES02)}
El vínculo por el que se recibe la PDU no tiene capacidad multiacceso (!C04). Existe una interfaz en dirección al core del grupo (C07), y la interfaz al nodo upstream es diferente a la interfaz de arribo (C12). En este caso se crea una entrada transitoria para el grupo, no leaf, (IA000), se activa el timer para eliminación de la entrada (IA040) (T03), y se agrega la interfaz de arribo como child para el grupo transitorio (IA002). Se desactiva el timer de envío de QUIT_NOTIFICATION asociado a la interfaz parent y al grupo, si está activo (IA071) (T06). Como evento externo, se genera un JOIN_REQUEST direccionado a la interfaz parent transitoria para el grupo. Se pasa a estado transitorio (ES02).

P17: C04 & !C05 & C10 & C07 & C12 {(IA000, IA040, IA002, IA071), (OE02), (ES02)}
El vínculo por el que se recibe la PDU tiene capacidad multiacceso (C04) no multicast (!C05). El nodo es DR en el vínculo (C10). Existe una interfaz en dirección al core del grupo (C07), y la interfaz al nodo upstream es diferente de la interfaz de arribo (C12). En este caso se crea una entrada transitoria para el grupo, no leaf, (IA000), se activa el timer para borrado de la entrada (IA040) (T03), y se agrega la interfaz de arribo como child para el grupo transitorio (IA002). Se desactiva el timer de envío de QUIT_NOTIFICATION asociado a la interfaz parent y al grupo, si está activo (IA071) (T06). Como evento externo, se genera un JOIN_REQUEST direccionado a la interfaz parent transitoria para el grupo. Se pasa a estado transitorio (ES02).

P18: C04 & !C05 & !C10 & C07 & C11 {(IA000, IA040, IA002, IA071), (OE02), (ES02)}
El vínculo por el que se recibe la PDU tiene capacidad multiacceso (C04) no multicast (!C05). El nodo no es DR en el vínculo (!C10). Existe una interfaz en dirección al core del grupo (C07), y la red asociada a la interfaz hacia el nodo upstream es diferente de la red asociada a la interfaz de arribo (C11). En este caso se crea una entrada transitoria para el grupo, no leaf, (IA000), se activa el timer para borrado de la entrada (IA040) (T03), y se agrega la interfaz de arribo como child para el grupo transitorio (IA002). Se desactiva el timer de envío de QUIT_NOTIFICATION asociado a la interfaz parent y al grupo, si está activo (IA071) (T06). Como evento externo, se genera un JOIN_REQUEST direccionado a la interfaz parent transitoria para el grupo. Se pasa a estado transitorio (ES02).

P19: C04 & C05 & C10 & C13 & C07 {(IA000, IA040, IA002, IA071), (OE02), (ES02)}
El vínculo por el que se recibe la PDU tiene capacidad multiacceso (C04) multicast (C05). El nodo es DR en el vínculo (C10) y la PDU ha sido enviada multicast (C13). Existe una interfaz en dirección al core del grupo (C07). En este caso se crea una entrada transitoria para el grupo, no leaf, (IA000), se activa el timer para borrado de la entrada (IA040) (T03), y se agrega la interfaz de arribo como child para el grupo transitorio (IA002). Se desactiva el timer de envío de QUIT_NOTIFICATION asociado a la interfaz parent y al grupo, si está activo (IA071) (T06). Como evento externo, se genera un JOIN_REQUEST direccionado a la interfaz parent transitoria para el grupo. Se pasa a estado transitorio (ES02).

P20: C04 & C05 & !C10 & !C13 & C07 & C12 {(IA000, IA040, IA002, IA071), (OE02), (ES02)}
El vínculo por el que se recibe la PDU tiene capacidad multiacceso (C04) multicast (C05). El nodo no es DR en el vínculo (!C10) y la PDU ha sido enviada unicast (!C13). Existe una interfaz en dirección al core del grupo (C07) y la red asociada a ella (parent) es diferente a la red asociada al vínculo de arribo (C12). En este caso se crea una entrada transitoria para el grupo, no leaf, (IA000), se activa el timer para borrado de la entrada (IA040) (T03), y se agrega la interfaz de arribo como child para el grupo transitorio (IA002). Se desactiva el timer de envío de QUIT_NOTIFICATION asociado a la interfaz parent y al grupo, si está activo (IA071) (T06). Como evento externo, se genera un JOIN_REQUEST direccionado a la interfaz parent transitoria para el grupo. Se pasa a estado transitorio (ES02).

Recepción de JOIN_ACK en estado transitorio

P21: C01 & C03 {(IA010, IA090, IA080, IA031, IA021, *C:(IA001, IA012)),(*C: OE01), (ES01)}

La PDU es recibida por la interfaz parent (transitoria) para el grupo (C01), y la entrada es leaf (C03); se crea la entrada on-tree para el grupo (IA010) activando el timer de borrado de la entrada (IA090) (T08) y el timer para realizar el echo request al parent (IA080) (T07). Cada interfaz child transitoria, es eliminada de la entrada transitoria (IA001) y agregada en la entrada on-tree (IA012). Finalmente, se desactivan los timers para la entrada transitoria leaf: el de expiración de la entrada (IA031) (T02) y el utilizado para la retransmisión de JOIN_REQUESTs (IA021) (T01). Como evento externo, se genera un JOIN_ACK por cada interfaz child (no local). Se pasa a estado on-tree (ES01).

P22: C01 & !C03 {(IA010, IA090, IA080, IA041, *C: (IA001, IA012)), (*C:OE01), (ES01)}
La PDU es recibida por la interfaz parent (transitoria) para el grupo (C01), y la entrada transitoria no es leaf (!C03). Se crea la entrada on-tree para el grupo (IA010) activando el timer de borrado de la entrada (IA090) (T08) y el timer para realizar el echo request al parent (IA080) (T07). Cada interfaz child transitoria, es eliminada de la entrada transitoria (IA001) y agregada en la entrada on-tree (IA012). Finalmente se desactiva el timer para borrado de la entrada no leaf (IA041) (T03). Como evento externo, se genera un JOIN_ACK por cada interfaz child (no local). Se pasa a estado on-tree (ES01).

Recepción de QUIT_NOTIFICATION en estado on-tree

P23: C02 & C04 & C05 & C06 {(IA050), ,(ES01)}
La PDU es recibida a través de una interfaz child para el grupo (C02) cuyo vínculo asociado es multiacceso (C04) con capacidad multicast (C05). La interfaz de arribo es la única child (C06). Se activa el timer para borrado de la entrada (IA050) (T04) (si ningún otro nodo child solicita permanecer en el grupo enviando un JOIN_REQUEST, la entrada será eliminada). No se generan eventos externos ni se cambia de estado.

P24: C02 & !(C04 & C05) & !C06 {(IA013), ,(ES01)}
La PDU es recibida por una interfaz child para el grupo (C02) cuyo vínculo asociado no cumple con la condición de multiacceso con capacidad multicast (!(C04 & C05)). La interfaz no es la única child para el grupo (!C06). En este caso, la interfaz child es eliminada (IA013). No se generan eventos externos ni se cambia de estado.

P25: C02 & !(C04 & C05) & C06 {(IA013, IA011, IA091, IA081, IA070, IA201), (OE03), (ES03)}
La PDU es recibida a través de una interfaz child (C02), cuyo vínculo asociado no es multiacceso con capacidad multicast (!(C04 & C05)); esta interfaz es la única child para el grupo (C06). En este caso, debe eliminarse la entrada child (IA013), la entrada (on-tree) para el grupo (IA011), desactivar el timer de borrado de la entrada on-tree (IA091) (T08), y el timer para solicitar eco (IA081) (T07). Además se llevan a cabo acciones relativas al envío del QUIT_NOTIFICATION al nodo parent: activación del timer para reenvío del QUIT_NOTIFICATION (IA070) (T06) e inicialización del contador de reintentos (IA201). Como evento externo, se genera el primer QUIT_NOTIFICATION direccionado al parent del grupo (OE03). Se pasa a estado sin entrada (ES03).

P26: C01 & C04 & C05 {(IA060), ,(ES01)}
La PDU es recibida a través de la interfaz parent para el grupo (C01). Su vínculo asociado tiene capacidad multiacceso (C04) multicast (C05). Esto significa que algún otro nodo child solicita

la desconexión del grupo; en este caso, se activa el timer de envío de JOIN_REQUEST (IA060) (T05) para evitar el podado del árbol. No se generan eventos externos ni se cambia de estado.

Recepción de FLUSH_TREE en estado on-tree

P27: C01 & !(C04 & C05) {((*C: IA013), IA011, IA081, IA091), ((*C: OE04)), (ES03)}
La PDU es recibida por la interfaz parent para el grupo (C01); el vínculo asociado no presenta la característica multiacceso multicast (!(C04 & C05)). En este caso, se eliminan las entradas child para el grupo (IA013), se elimina la entrada para el grupo (IA011), se desactiva el timer para realizar el ECHO_REQUEST (IA081) (T07) y el timer para eliminación de la entrada on-tree (IA091) (T08). Como evento externo se genera un FLUSH_TREE por cada interfaz child (OE04). Se pasa a estado sin entrada (ES03).

P28: C01 & C04 & C05 {((*C: IA013), IA011, IA081, IA091, IA061), ((*C:OE04)), (ES03)}
La PDU es recibida por la interfaz parent para el grupo (C01); el vínculo asociado posee capacidad multiacceso (C04) multicast (C05). En este caso, se eliminan las entradas child para el grupo (IA013), se elimina la entrada para el grupo (IA011), se desactiva el timer para realizar el ECHO_REQUEST (IA081) (T07), el timer para eliminación de la entrada on-tree (IA091) (T08), y el timer para envío de JOIN_REQUEST, en el caso de que estuviera activo (IA061) (T05). Como evento externo se genera un FLUSH_TREE por cada interfaz child. Se pasa a estado sin entrada (ES03).

Recepción de ECHO_REQUEST en estado on-tree

P29: C02 {(IA100), , (ES01)}
La PDU es recibida por una interfaz child (C02). En caso de que el timer para envío del ECHO_REPLY correspondiente esté inactivo, es activado (IA100) (T09). No se generan eventos externos ni se cambia de estado.

P30: C01 & C04 & C05 {(IA081), , (ES01)}
La PDU es recibida por la interfaz parent para el grupo (C01) y su vínculo asociado presenta la característica multiacceso (C04) multicast (C05). En este caso, se desactiva el timer para envío de ECHO_REQUEST al parent (IA081) (T07) si estaba activo. No se generan eventos externos ni se cambia de estado.

Recepción de ECHO_REPLY en estado on-tree

P31: C01 {(IA092), , (ES01)}
La PDU es recibida por la interfaz parent para el grupo (C01). En este caso se reactiva el timer de eliminación de la entrada (IA092) (T08), cargándolo con su valor original. No se generan eventos externos ni se cambia de estado.

Recepción de una solicitud JOIN-GROUP en estado on-tree

P32: V {(IA012), , (ES01)}
En este caso se agrega la interfaz (local) como child para el grupo (IA012). No se generan eventos externos ni se cambia de estado.

Recepción de una solicitud JOIN-GROUP en estado transitorio

P33: V {(IA002), , (ES01)}
En este caso se agrega la interfaz (local) como child en la entrada transitoria para el grupo (IA002). No se generan eventos externos ni se cambia de estado.

Recepción de una solicitud JOIN-GROUP en estado sin entrada

P34: C07 {(IA000, IA020, IA030, IA002), (OE02), (ES02)}
Si existe una interfaz en dirección al core del grupo (posiblemente a través de nodos intermedios) (C07), se crea una entrada transitoria, leaf (IA000), y se activan los timers correspondientes: para reenvío de JOIN_REQUESTs (IA020) (T01) y para eliminación de la entrada transitoria (IA030) (T02). Luego se agrega la interfaz local como child para el grupo (IA002). El evento externo consiste en generar un JOIN_REQUEST direccionado al parent (transitorio) del grupo. Se pasa a estado transitorio.

Recepción de una solicitud LEAVE-GROUP en estado on-tree

P35: !C06 {(IA013), , (ES01)}
Si la interfaz local no es la única interfaz child para el grupo (!C06), se elimina la interfaz child (IA013). No se generan eventos externos ni se cambia de estado.

P36: C06 & !(C08 & C09) {(IA013, IA011, IA081, IA091, IA070), (OE03), (ES03)}
Si la interfaz local es la única child para el grupo (C06) y el vínculo asociado a la interfaz parent para el grupo no tiene la característica multiacceso multicast (!(C08 & C09)), se elimina la interfaz child (IA013), se elimina la entrada para el grupo (IA011), se desactiva el timer para envío de ECHO_REQUEST al parent (IA081) (T07), se desactiva el timer para eliminación de la entrada permanente (IA091) (T08) y se activa el timer para reenvío de QUIT_NOTIFICATION al parent (IA070) (T06). Como evento externo, se genera un QUIT_NOTIFICATION al parent (OE03). Se pasa a estado sin entrada.

P37: C06 & C08 & C09 {(IA013, IA011, IA081, IA091, IA070, IA061), (OE03), (ES03)}
Si la interfaz local es la única child para el grupo (C06) y la interfaz parent es multiacceso (C08) multicast (C09), se elimina la interfaz child (IA013), se elimina la entrada para el grupo (IA011), se desactiva el timer para envío de ECHO_REQUEST al parent (IA081) (T07), se desactiva el timer para la eliminación de la entrada permanente (IA091) (T08) y se activa el timer para reenvío de QUIT_NOTIFICATION al parent (IA070) (T06). En caso de que el timer para envío de JOIN_REQUEST (T05) esté activo, se lo desactiva (IA061). Como evento externo, se genera el primer QUIT_NOTIFICATION al parent (OE03). Se pasa a estado sin entrada.

Retransmit_Join_Request_Timer (T01): timeout en estado transitorio

P38: V {(IA020), (OE02), (ES02)}
Este evento producirá el reenvío del JOIN_REQUEST sobre la interfaz parent transitoria para el grupo (OE02) y la activación del timer (IA020) (T01). Sólo se produce en entradas leaf.

Delete_Leaf_Transient_Entry_Timer (T02): timeout en estado transitorio

P39: V { ((*C:IA003), IA001, IA021) , , (ES03)}
Este evento, producido si después de un cierto tiempo de enviado el primer JOIN_REQUEST no se obtiene respuesta del supuesto parent, producirá la eliminación de las entradas child transitorias (IA003), la eliminación de la entrada transitoria leaf (IA001) y la desactivación del timer para retransmisiones de JOIN_REQUEST (IA021) (T01). No se generan eventos externos. El nuevo estado es sin entrada.

Delete_Transient_Entry_Timer (T03): timeout en estado transitorio

P40: V { ((*C:IA003), IA001), , (ES03)}
Este evento, producido si luego de un cierto tiempo de enviado el JOIN_REQUEST no se obtiene respuesta del supuesto parent, producirá la eliminación de las entradas child transitorias (IA003), de la entrada transitoria no leaf (IA001). No se generan eventos externos. El nuevo estado es sin entrada.

Cache_Deletion_Timer (T04): timeout en estado on-tree

P41: !C06 {(IA013), (OE04), (ES01)}
Este evento sólo puede producirse relativo a una interfaz child asociada a un vínculo multiacceso multicast. Si la interfaz child correspondiente al timer que produce este evento no es la única child para el grupo (!C06), se elimina dicha entrada child (IA013). Se genera un FLUSH_TREE (OE04) sobre la interfaz child. No se cambia de estado.

P42: C06 {(IA013, IA081, IA091, IA011, IA070), (OE04, OE04), (ES03)
Este evento sólo puede producirse relativo a una interfaz child asociada a un vínculo multiacceso multicast Si la interfaz child correspondiente al timer que produce el evento es la única child para el grupo (C06), se elimina dicha interfaz child (IA013), y luego se desactiva el timer para solicitud de eco correspondiente a la entrada para el grupo (IA081) (T07), y el timer para eliminación de la entrada (IA091) (T08); se elimina la entrada on-tree (IA011), y se activa el timer para enviar QUIT_NOTIFICATION al parent (IA070) (T06). Como eventos externos, se envía un QUIT_NOTIFICATION (el primero) al parent (OE03), y un FLUSH_TREE (OE04) direccionado a la interfaz child. Se pasa a estado sin entrada (ES03).

Transmit_Join_Request_Timer (T05): timeout en estado on-tree

P43: V { , (OE02), (ES01)}

Este evento indica que deben tomarse acciones para evitar que el parent elimine la entrada multicast. No hay acción interna. Se genera un evento externo, consistente en el envío de un JOIN_REQUEST (OE02) al parent. No se cambia de estado.

Quit_Notification_Timer (T06): timeout en estado sin entrada

P44: C14 {(IA200, IA070), (OE03), (ES03)}
Si la variable de conteo de reenvíos es menor que el límite (C14), se aumenta en uno dicha variable (IA200) y se activa el timer para producir el próximo envío (T06) (IA070). Se genera un evento externo consistente en el envío de un QUIT_NOTIFICATION (OE03) al parent.

P45: !C14 { , (OE03), (ES03)}
Si la variable de conteo de reenvíos es igual o mayor que el límite (!C14), no se realizan acciones internas. Se envía un QUIT_NOTIFICATION (OE03) (el último) al parent. No se cambia de estado.

Echo_Request_Timer (T07): timeout en estado on-tree

P46: V {(IA080), (OE05), (ES01)}
Al producirse este evento, se activa nuevamente el timer (T07) (IA080). Se genera un ECHO_REQUEST (OE05) y se envía al parent. No se produce cambio de estado.

Delete_on-tree_Timer (T08): timeout en estado on-tree

P47: !(C04 & C05) {(*C:(IA013, IA081), IA011, IA070), (*C:(OE04), OE03), (ES03)}
Si el vínculo asociado a la interfaz parent del grupo no tiene capacidad multiacceso multicast (!(C04 & C05), se eliminan todas las entradas child (IA013), y se desactiva su timer para ECHO_REQUEST (T07) (IA081). Por último se elimina la entrada para el grupo (IA011) y se activa el timer para enviar QUIT_NOTIFICATION al parent (IA070) (T06). Como eventos externos, se envía un QUIT_NOTIFICATION (el primero) al parent (OE03), y un FLUSH_TREE (OE04) direccionado a cada interfaz child. Se pasa a estado sin entrada (ES03).

P48: C04 & C05 & C10 {(*C:(IA013, IA081), IA011, IA070), (*C:(OE04), OE03), (ES03)}
Si el vínculo asociado a la interfaz parent del grupo tiene capacidad multiacceso (C04) multicast (C05) y el nodo es DR en el vínculo (C10) se eliminan todas las entradas child (IA013), y se desactiva su timer para ECHO_REQUEST (T07) (IA081). Por último se elimina la entrada para el grupo (IA011) y se activa el timer para enviar QUIT_NOTIFICATION al parent (IA070) (T06). Como eventos externos, se envía un QUIT_NOTIFICATION (el primero) al parent (OE03), y un FLUSH_TREE (OE04) direccionado a cada interfaz child. Se pasa a estado sin entrada (ES03).

P49: C04 & C05 & !C10 {(*C:(IA013, IA081, IA051), IA011, IA070), (*C:(OE04), OE03), (ES03)}
Si el vínculo asociado a la interfaz parent del grupo tiene capacidad multiacceso (C04) multicast (C05) y el nodo no es DR en el vínculo (!C10), se eliminan las entradas child para el grupo (IA013), se desactiva el timer para ECHO_REQUEST (T07) (IA081) y el timer para reenvío de

JOIN_REQUEST (T05) (IA051) si estaba activo. Por último se elimina la entrada para el grupo (IA011) y se activa el timer para enviar QUIT_NOTIFICATION al parent (IA070) (T06). Como eventos externos, se envía un QUIT_NOTIFICATION (el primero) al parent (OE03), y un FLUSH_TREE (OE04) direccionado a la interfaz child. Se pasa a estado sin entrada (ES03).

Echo_Reply_Timer (T09): timeout en estado on-tree

P50: V {, (OE06), (ES01)}
Este evento ocurrirá asociado a la interfaz parent para el grupo. No se producen acciones internas. Como evento de salida, se envía un ECHO_REPLY (OE06) por la interfaz parent. No se cambia de estado.

5.3 Resumen

En este capítulo se presentó una descripción formal de la operación del protocolo CBT versión 2 en forma de máquina de estados finitos. La especificación realizada tiene por objeto ser utilizada como base para desarrollar la implementación, en este caso en el ambiente de simulación Ns. No se intenta en modo alguno la derivación del código en forma automática a partir de la especificación realizada, ni su utilización en aspectos tales como comprobación de la correctitud del protocolo, para lo cual sería necesario su especificación en un lenguaje apropiado, como por ejemplo SDL.