

Analysis of the Dynamic Service Flow Management Transactions Protocol for MAC IEEE 802.16. An Approach

Ana Morales Bezeira, María Villapol, Eugenio Scalise, Cesar Naranjo y Fabian Álvarez

Resumen— In this work the WiMax MAC Protocol for Dynamic Service Flow management and their transactions is analyzed by showing this several pitfalls of the specification and how this can affect the implementation of the protocol using Colored Petri Nets (CPNs). A detailed description of the protocol is developed at level of transactions it supports, and at level of service flows. The modeling and analysis of the protocol are important to improve the current specification and to support the reuse of the concepts in emerging networks. We show the potentialities of using a transaction oriented approach in the description of this kind of protocols. The contributions of this paper are focused on providing a clear description of the protocol, presentation of the model, validation and analysis of the model according to the generation of the Dynamic Service Flow management transactions protocol language, as well as the determination of regular expressions Language.

Palabras Clave— Dynamic Service Flow, Colored Petri Nets, Pitfalls, WiMax MAC IEEE 802.16 Protocol, Transactions, Protocol Language.

I. INTRODUCCIÓN

EL constante incremento en el número de dispositivos móviles y los requerimientos de las aplicaciones son factores determinantes de las características de los futuros sistemas 5G [1]. El conjunto de requerimientos y los retos a los que se enfrentan las futuras redes en los sistemas 5G incluyen: mayores tasas de datos, soporte a mayor número de dispositivos de forma simultánea, altos volúmenes de tráfico, menor latencia, mejoras en la cobertura tanto en espacios internos (*indoor*) como externos (*outdoor*), bajo consumo de energía, mayor asimetría en el tráfico y alta confiabilidad [1].

Sometido para su revisión el 22 de abril de 2017.

Ana Morales Bezeira pertenece al Laboratorio de Redes Móviles, Inalámbricas y Sistemas Distribuidos (ICARO), Universidad Central de Venezuela, Caracas, Venezuela (e-mail: ana.morales@ciens.ucv.ve).

María Villapol pertenece a School of Engineering, Computer and Mathematical Sciences, Auckland, New Zealand (e-mail: mariaelena.villapol@gmail.com).

Eugenio Scalise pertenece al Centro de Investigación de Ingeniería del Software y Sistemas, Universidad Central de Venezuela, Caracas, Venezuela (e-mail: eugenio.scalise@ciens.ucv.ve).

Cesar Naranjo Álvarez pertenece al Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador (e-mail: canaranjo@espe.edu.ec).

Fabian Álvarez pertenece al Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE, Latacunga, Ecuador (e-mail: faalvarez@espe.edu.ec).

Estos requerimientos demandan completa movilidad y alta conectividad. Por lo tanto, el producto final será un conjunto de estándares que reúnan el resultado de una vasta cantidad de investigaciones relacionadas a las tecnologías existentes y emergentes en un período de varios años [1].

En este sentido, tales tecnologías deben estar adecuadamente soportadas en su núcleo por protocolos de comunicación claramente definidos y especificados.

Un protocolo de comunicación orientado a conexiones tiene como principal objetivo la reservación de recursos. Este tipo de protocolo debe poder crear, mantener y liberar las conexiones apropiadas, para un flujo de datos en todas las entidades que intervienen a lo largo de la ruta de transporte de los mismos.

Las actividades de reservación de recursos en redes con condiciones de tráfico y rutas altamente cambiantes, tales como las redes IP y las redes móviles e inalámbricas, no resultan ser tareas fáciles, ya que los requerimientos de QoS desde las capas superiores para un flujo de datos pueden variar con el tiempo, y tales cambios deben ser soportados y almacenados en todos los nodos que conforman la red. Así mismo, estos cambios deben ser refrescados cada vez que las características de QoS del tráfico sean modificadas o el flujo de tráfico sea eliminado. Tales actividades pueden consumir mucho tiempo de procesamiento de los nodos de la red, y cierta información podría permanecer almacenada en los dispositivos intermedios aunque ya no formen parte del flujo de datos actual.

Un protocolo para la gestión de reservación de recursos en la forma de Flujo de Servicio (SF, *Service Flow*) en redes móviles e inalámbricas es el Protocolo WiMax MAC (*Medium Access Control*) para la gestión de Flujos de Servicio Dinámico (DSF, *Dynamic Service Flows*) [2], [3]. Este protocolo consta de un conjunto de mensajes y procedimientos que permiten a una estación base (BS, *Base Station*) y una estación suscriptora (SS, *Station Subscriber*) la creación, el mantenimiento y la liberación de SF para el transporte de datos en una conexión entre dos entidades WiMax.

Un SF representa el principal componente en el protocolo de la capa MAC para la gestión de las conexiones en el transporte de datos. Este protocolo es orientado a la conexión, y proporciona a la BS el control sobre los DSF que requieran ser establecidos entre los dispositivos de la red (SS o MS, *Mobile Subscriber*). El protocolo utiliza un enfoque del tipo *soft-state* para lidiar con las características cambiantes de los SF, aspecto altamente deseable en los entornos móviles.

El protocolo de la capa MAC para la gestión de los DSF complejo, debido al conjunto de aspectos que involucran las actividades de creación, mantenimiento y liberación de los DSF. En la especificación [2], [3], el protocolo se describe utilizando narrativas, diagramas de flujos de estado, diagramas de secuencias de caracteres y diagramas de flujo. Así mismo, se introduce la noción de transacciones para las actividades de gestión de los DSF. Las transacciones facilitan la especificación del protocolo, así como también la definición jerárquica de los procedimientos asociados a éstas.

La utilización de múltiples artefactos como los descritos previamente hacen que el núcleo del protocolo en la MAC resulte en procesos y procedimientos sumamente complejos de implementar, llegándose a encontrar imprecisiones, ambigüedades e incluso vacíos de información.

En el contexto de los requerimientos de QoS de las redes 5G, pensadas para soportar la creciente demanda de servicios multimedia, y diferentes tipos de tráfico simultáneo en los dispositivos, es donde protocolos como el desarrollado en la MAC de WiMax es altamente probable y deseable que se utilice.

En este sentido, un análisis del protocolo MAC para la gestión de los DSFs resulta de suma importancia, para facilitar la reutilización de los conceptos presentes en la especificación del mismo en las futuras redes, y también para mejorar la descripción actual de los procedimientos existentes.

Las actividades de descripción y análisis de protocolos ya han sido llevadas a cabo en trabajos previos, utilizando para ello métodos formales como lo son las Redes de Petri Coloreadas (CPNs, *Coloured Petri Nets*) [4], [5]. Algunos de tales trabajos son: el análisis y validación del Protocolo para la Reservación de Recursos (RSVP, *Resource Reservation Protocol*) en el cual el autor utiliza las CPNs para validar el protocolo [6]. Tal validación fue llevada a cabo utilizando la metodología para la validación y verificación de protocolos propuesta por Billington [7].

En [8], las CPNs son utilizadas para proporcionar una mejora a la especificación del protocolo que soporta el mecanismo de Solicitud Reconocimiento Automáticos en WiMax (*ARQ, Automatic Repeat Request*); en este trabajo los autores realizan una descripción funcional del protocolo, encontrando ciertos inconvenientes, y proponiendo mejoras a los procedimientos internos de la especificación.

En [9], [10] y [11] se lleva a cabo la revisión y validación formal de manera incremental de la Especificación del Servicio del protocolo MAC para la gestión de las conexiones en IEEE 802.16, abarcando la generación del lenguaje para la Especificación del Servicio. Cabe destacar que los trabajos desarrollados en [9], [10] y [11] no estuvieron enfocados en estudiar, analizar y validar los protocolos asociados a los procedimientos para la gestión de los DSF en WiMax, así mismo no se han hallado trabajos hasta el momento publicados sobre el estudio de tales procedimientos. Los trabajos publicados relacionados a WiMax, se centran en mejoras en movilidad, ancho de banda, mejoramiento del uso del espectro, entre otros [12], [13], [14].

En este trabajo, el protocolo MAC para la gestión de los

DSF es modelado y analizado, con el objeto de proveer una mejor descripción de la especificación del mismo que la dada en los documentos oficiales del estándar [2], [3] y para mostrar los inconvenientes y debilidades encontrados en la especificación del estándar, y cómo éstos podrían afectar las implementaciones del protocolo. También, se analizan las potencialidades de utilizar el enfoque orientado a transacciones en la descripción de protocolos para la reservación de recursos. En este sentido, se modela y valida la el protocolo MAC IEEE 802.16 para gestión de los DSF y sus transacciones, utilizando para ello las CPNs [4], [5].

Entre las contribuciones resaltantes de este trabajo se encuentran: la descripción detallada y precisa del protocolo MAC para la gestión de los DSF, el desarrollo del modelo formal, la validación del modelo a través de la determinación de las expresiones regulares del lenguaje de los FSA obtenidos, lo cual facilitó la generación del lenguaje del protocolo, y mostrando los diferentes inconvenientes encontrados en la especificación.

El artículo está distribuido de la siguiente manera. En la Sección 2, se describe la metodología empleada. En la Sección 3, se presenta una descripción detallada del protocolo MAC para la gestión de los DSF. Seguidamente en la sección 4, se presenta y describe el modelo desarrollado. En la sección 5 se presenta el análisis del protocolo utilizando para ello el análisis del Grafo de Ocurrencias proporcionado por las CPNs, y su posterior análisis y generación del lenguaje, utilizando para ello métodos de minimización y generación de máquinas de estado finito finitas (FSA, *Finite State Automaton*). Finalmente, en la Sección 6 se presentan las conclusiones.

II. METODOLOGÍA

Con el objetivo de analizar el protocolo MAC para la gestión de los DSF se realizan las siguientes actividades:

- 1) *Especificación del protocolo MAC para la gestión de los flujos de servicios dinámicos.* Descripción del protocolo usando, narrativa y diagramas de transición estado con el objetivo de proveer un claro entendimiento del mismo.
- 2) *Modelado y simulación del protocolo MAC para la gestión de los flujos de servicios dinámico.* Se modela el protocolo basado en la especificación del mismo utilizando para ello las CPNs, con la finalidad de resaltar inconvenientes observados en dicha especificación. Las simulaciones fueron utilizadas para eliminar errores iniciales del modelo. Por ejemplo, para encontrar errores tales como secuencias de eventos del protocolo erróneas.
- 3) *Análisis.* El análisis consistió en validar el protocolo según las propiedades dinámicas de las CPNs. En este trabajo se utiliza el método basado en el OG. Luego se aplican técnicas de reducción del OG a máquinas de estado finito con la ayuda del algoritmo desarrollado en [7] e implementado en [16], con la finalidad de generar y comparar la secuencia de ocurrencia de las transacciones (es decir, lenguaje del FSA de las transacciones) y la secuencia de ocurrencia de los cambios de estado de los DSF (es decir, lenguaje del FSA de los DSF).

III. PROTOCOLO PARA LA GESTIÓN DE FLUJOS DE SERVICIOS DINÁMICOS DE LA MAC IEEE 802.16

La tecnología WiMax es una tecnología de acceso inalámbrico de banda ancha para redes de área metropolitana, soportada en el estándar IEEE 802.16 [3], [4]. La tecnología proporciona diferentes niveles de QoS a través de un conjunto de mecanismos implementados en la capa MAC, los cuales permiten garantizar los servicios de voz, datos y video.

Para proporcionar tales garantías, el estándar IEEE 802.16 define un conjunto de protocolos que proveen las funcionalidades bases del sistema de acceso a la red, asignación de anchos de banda asociados a las conexiones de transporte de datos establecidas entre dispositivos cliente (SS o MS), y una estación BS. En este sentido, cada conexión debe estar asociada con un servicio de datos. Este servicio de datos está asociado a un flujo de servicio (SF, *Service Flow*) que es provisto a través de un conjunto de transacciones que permiten gestionar las conexiones de datos. Así, tales transacciones son soportadas por un protocolo de gestión a través del intercambio de mensajes del tipo DSF.

A. Flujos de Servicios y Transacciones Asociadas

Un SF es un servicio de transporte MAC que proporciona transporte unidireccional de paquetes, transmitidos desde una estación suscriptora a una estación base (enlace *Uplink*), o desde la estación base a una estación suscriptora (enlace *DownLink*). Está identificado de forma única por un ID de 32 Bits, denominado Identificador de Flujo de Servicio (SFID, *Service Flow Identifier*) [3], [4]. Un SF se caracteriza por un conjunto de parámetros de QoS.

B. Tipos de Flujos de Servicio

El estándar define tres tipos básicos de SF [4]: Flujos de Servicios Provistos, Flujos de Servicio Admitidos y Flujos de Servicio Activos. Los Flujos de Servicios Provistos son el conjunto de SFs que pueden ser soportados por la red, sin embargo, éstos aún no han sido admitidos, ni activados por parte de la estación base. La red se encargará de asignar un SFID a tales SF.

Los Flujos de Servicios Admitidos son el conjunto de SFs a los cuales la estación base le asigna recursos (es decir que su conjunto de parámetros de QoS han sido admitidos, pero tales recursos aún no han sido activados). Las estaciones suscriptoras podrán solicitar la admisión de un SF, utilizando el envío de un mensaje de gestión denominado **Solicitud de Adición de Servicio Dinámico** (DSA-REQ, *Dynamic Service Addition Request Message*), llevando consigo los parámetros de SFID y de QoS requeridos. El conjunto de parámetros QoS admitidos y no activados aún, puede ser dinámicamente modificado por la estación base, a través del uso de un mensaje del tipo **Mensaje de Solicitud de Cambios de Servicio Dinámico** (DSC-REQ, *Dynamic Service Change Request Message*).

Por su parte, los Flujos de Servicios Activos son el conjunto de SF que tiene un conjunto de parámetros QoS activos. Para tales flujos, es posible solicitar y conceder ancho de banda para el transporte de paquetes de datos.

Cuando una estación suscriptora o una estación base inicia

un procedimiento de establecimiento de una conexión de transporte de datos, la conexión se establecerá sólo para los SF Admitidos y Activos. Por lo tanto, a este SF, identificado por un SFID, se le asociará un único Identificador de Conexión (CID, *Connetion Identifier*), tal CID es de 16 bits, con lo cual un Flujo de Servicio Activo estará compuesto de un SFID+CID. El CID es utilizado para la entrega de los datos del usuario a las interfaces de la capa MAC [4].

C. Protocolo para la Gestión de Flujos de Servicios Dinámicos y sus Transacciones Asociadas

Los DSF pueden ser dinámicamente creados, modificados o eliminados. Para esto, el estándar provee un conjunto de mensajes de gestión MAC denominados DSA, DSC y DSD, los cuales representan las transacciones de gestión asociadas a los DSF [3], [4]. Como se indicó anteriormente, los mensajes del tipo DSA y DSC permiten crear y modificar un DSF, mientras que el mensaje del tipo **Solicitud de Eliminación de Servicio Dinámico** (DSD-REQ, *Dynamic Service Deleted*), permite la eliminación de un DSF existente. Esto conllevará a la liberación de los recursos de red asignados a la conexión para el SF.

El protocolo es descrito utilizando una visión jerárquica, que muestra su comportamiento en diferentes niveles de abstracción, a continuación se describe el protocolo.

D. Visión Jerárquica del Protocolo MAC para la Gestión DSF

1) Nivel Superior

La Fig. 1 muestra la máquina de estado para las transacciones de los DSF en el nivel más alto de abstracción. En ella se definen dos estados asociados a las transacciones que ocurren en los dispositivos que implementan la MAC WiMax (estaciones suscriptoras y estaciones base), y que requieren establecer una conexión. El estado *Null*, es aquel donde no existe ninguna conexión establecida para un DSF entre dispositivos. Un dispositivo que desee establecer una conexión de para un SF solicitará el establecimiento de un DSF, esto ocurre a través de una transacción enviando un mensaje del tipo DSA. Una vez creado, admitido y activado el DSF, este pasa al estado *Operational*, y se le asignará un SFID+CID, y permanecerá en este estado mientras ocurran transacciones del tipo DSC. Un DSF sólo podrá volver al estado *Null*, si se dispara una transacción del tipo DSD.

Cada secuencia de mensajes DSx representa una única transacción, con un único *TransactionID* asociado. Las transacciones DSA y DSC, consisten de una secuencia de mensajes del tipo: solicitud (REQ)/ respuesta (RSP)/ reconocimiento (ACK).

Mientras que la transacción DSD consiste únicamente de una secuencia del tipo: solicitud (REQ) / respuesta (RSP).

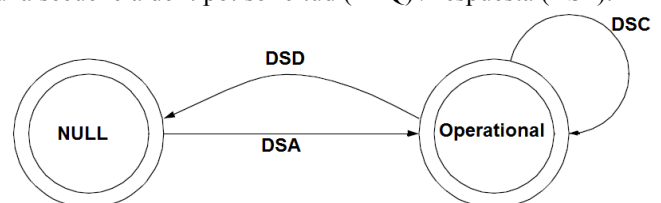


Fig. 1. Máquina de estado para las Transacciones de los Flujos de Servicios Dinámicos [4].

2) Segundo Nivel

Los DSF son creados según las necesidades de las entidades SS o BS. La Fig. 2 muestra en más detalle el comportamiento que debe tener el protocolo, basado en los cambios de estado de los DSF en las entidades involucradas. En ella, se definen 8 posibles sub-estados por los cuales puede pasar un DSF, ellos son: *AddinLocal*, *AddingRemote*, *AddFailed*, *Nominal*, *ChangingLocal*, *ChangingRemote*, *DeletingDeleted*.

El estado *Operational* de la Fig. 1 es representado en este nivel por estos ocho sub-estados a medida que ocurren las transacciones DSA y DSC. Una vez que el DSF se encuentre estable en el estado *Operational*, un DSF podrá permanecer iterando en un estado *Nominal*, trascendiendo por los múltiples sub-estados, tanto en la entidad iniciadora (entidad Local), como en la entidad remota.

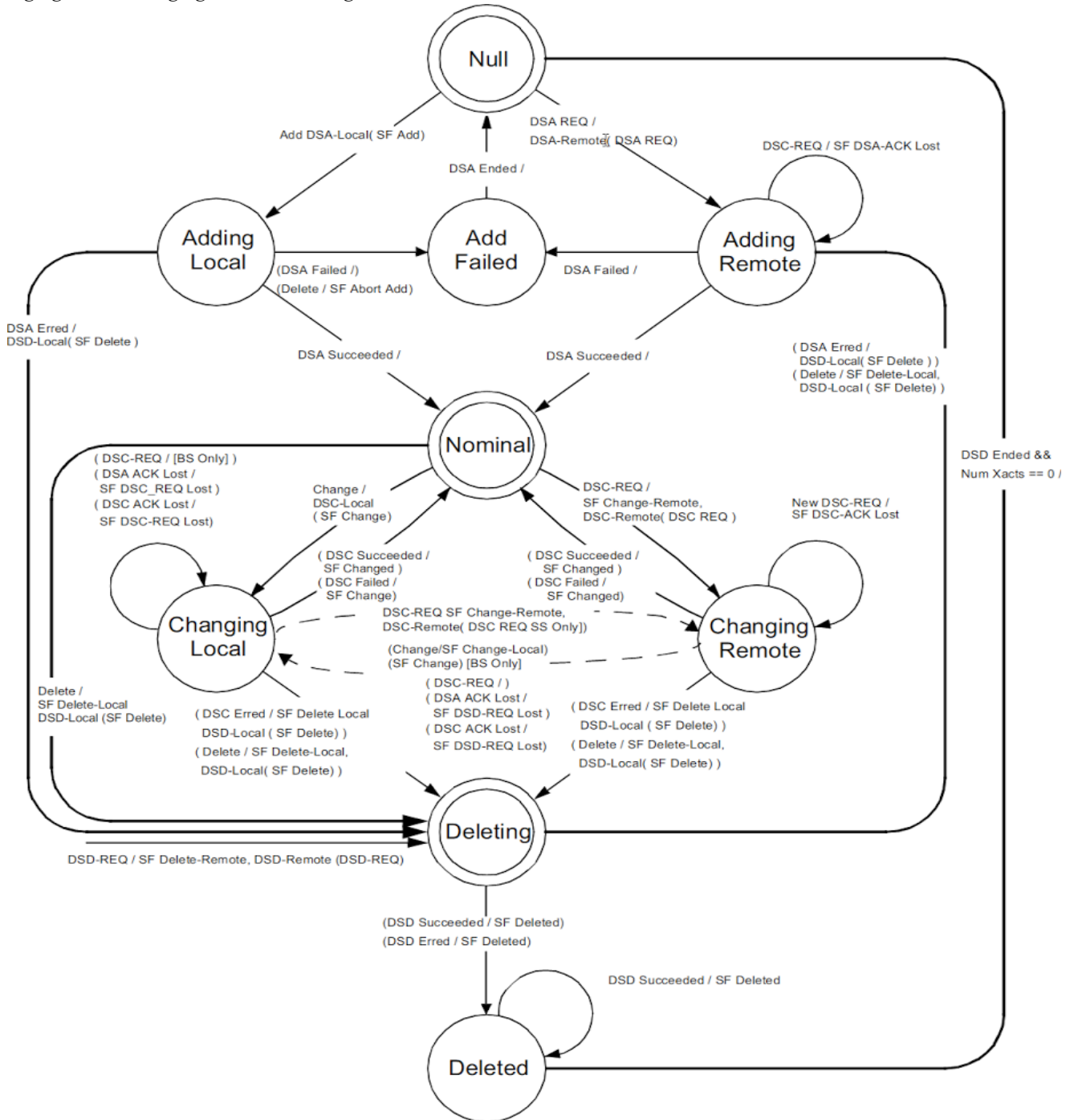


Fig. 2. Diagrama de Transición de Estados para Flujos de Servicios Dinámicos [4].

Es de destacar, que un DSF no retornará al estado *Null*, hasta que no haya sido borrado y todas las transacciones en curso (dentro del estado nominal) no hayan terminado. A continuación se presentan las entradas válidas para los

diagramas de transición de estado [4]
 Las entradas al diagrama de transición de estados de una transacción DSx, desde el diagrama de transición de estados para un DSF (Fig. 2) son las acciones: *SFAdd*, *SFChange*, *SF*

Delete, SFAbort Add, SFChange-Remote, SFDelete-Local, SF DeleteRemote, SF DSA-ACK Lost, SF DSC-REQ Lost, SF DSC-ACK Lost, SFChanged y S Deleted.

Para la transacción DSA, las entradas válidas son: *DSASucceeded, DSASFailed, DSA-ACKLost, DSAErred y DSAEnded*; para la transacción DSC, las entradas válidas son: *DSCSucceeded, DSCFailed, DSC-ACKLost, DSCErred y DSCEnded*; finalmente para la transacción DSD, las entradas válidas son: *DSDSucceeded, DSDErred y DSDEnded*.

Las transiciones de estado para un DSF desde una transacción son etiquetadas siguiendo la notación [4]: <causa de la transición>/<mensajes y eventos disparados por la transición>

Si hay múltiples eventos o mensajes antes del “/” separados por una coma, significa que cualquiera de éstos podría causar una transición. Así mismo, si hay múltiples eventos o mensajes listados después del “/”, significa que todas las acciones especificadas en el arco y separadas por comas deberán llevarse a cabo con la transición.

Las posibles causas de la transición de estado para un DSF desde una transacción, podrían ser la recepción de mensajes de transacción en la entidad remota (por ejemplo, DSA-REQ DSC-REQ); o la solicitud de una actividad de gestión en la entidad local desde capas superiores (por ejemplo, un *SF Add* o un *SF Change*).

Como un ejemplo de esta notación, véase en la Fig. 2 la siguiente inscripción en el arco que conecta al estado *Nominal* con el estado *ChangingRemote*: DSC-REQ/SFChange Remote, DSC-Remote (DSC-REQ). Esto debe leerse de la siguiente manera: una vez que en la entidad no iniciadora (entidad MAC remota) se ha recibido un mensaje del tipo DSC-REQ, este disparará el envío del evento *SF Change Remote*, a la transacción apropiada para el flujo de servicio indicado por el SFID en los parámetros del mensaje recibido, e iniciará la ejecución de la transacción. Igualmente pasará el evento DSC-Remote a la capa superior en la entidad remota.

Igualmente, la causa de una transición puede ser la ocurrencia de una transacción DSx. Cada transacción en la Fig. 2 sigue el siguiente notación [4]: DSx-[Local|Remote] (initial_input)

Donde DSx se refiere al tipo de transacción; [Local|Remote] identifica si la transacción es disparada localmente desde la entidad iniciadora de la transacción (BS o SS), o remotamente desde la entidad no iniciadora (BS o SS); initial_input identifica la entrada inicial que puede ser: un *SF Add, DSAREQ, SF Add, DSA-REQ, SF Change, DSC-REQ, SF Delete*, o *DSD-REQ*.

Un ejemplo de tal notación es la inscripción en el arco que conecta al estado *Null* con el estado *AddingLocal* de la Fig. 2: Add DSA-Local (SF Add). Ello se refiere, a que un DSF pasa al estado *AddingLocal*, en la entidad iniciadora (Local) debido a que en el nivel superior (nivel de transacciones, Fig. 1) ocurrió una transacción del tipo DSA, y disparó en la MAC de la entidad iniciadora un evento del tipo *SF Add*.

Cada transacción es representada en el estándar por un diagrama de transición de estados de la transacción [4], esto con el fin de proveer señalización entre las transacciones

DSA, DSC y DSD (es decir, con la finalidad de mantener sincronizadas las entidades que intentan establecer un DSF a través del intercambio de mensajes de las transacciones).

Así mismo, cada diagrama de transición de estados de la transacción se comunica únicamente con su diagrama de transición de estado del DSF al que corresponda en la Fig. 2. Este proceso se logra una vez que los flujos de servicio dinámicos son creados, modificados o eliminados como consecuencia de la ocurrencia de las transacciones. Así, el diagrama de transición de estados de la Fig. 2, filtra los mensajes DSx, y pasa éstos a la apropiada transacción.

E. Inconvenientes de la Especificación

Los diagramas presentados en la Fig. 1 y la Fig. 2, muestran un conjunto de inconsistencias y vacíos de información que pueden interpretarse como errores en la especificación.

1) No está claro el sentido que debe tener el arco que conecta al estado *Deleted* con el estado *Null* en la Fig. 2, ya que tal arco no tiene inicio o fin. Esto es evidenciado en una minuciosa revisión, ya que en ningunos de los documentos oficiales asociados a la especificación del estándar, se indica el sentido de tal arco [3], [4], [17], [18], [19]. Esto podría interpretarse como un vacío en la especificación, dejándose a la interpretación del fabricante la implementación de tal transición de estado. Esta situación también es observada en el arco que conecta a los estados *Adding Remote* y *Deleting*. En este trabajo se ha asumido que existe una transición desde el estado *Deleted* al estado *Null*, sólo si ocurre el evento "DSD Ended && Num Xacts==0", ya que es el comportamiento esperado en los protocolos de comunicación basados en establecimientos y liberaciones de conexiones.

2) El estado *Deleting* tiene un arco de entrada con la siguiente inscripción: DSD-REQ/SFDelete-Remote, DSD-Remote (DSD-REQ), para tal arco no se indica en la especificación cuál es el estado origen del mismo [3], [4].

Para los autores resulta confusa la manera como se relacionan y comunican las Fig. 1 y 2. Así mismo, es complejo comprender desde la Fig. 1, que todo lo que ocurre internamente en ella, luego de que se dispara un evento del tipo DSA, correspondería a todo lo que puede suceder internamente estando en el estado *Operational* de la Fig. 2. Por ejemplo, si la capa superior a la MAC desea establecer un DSF, en la entidad iniciadora se disparará un procedimiento del tipo *SF Add* (entidad local, ver Fig. 2), que llevará al DSF del estado *Null* al sub-estado *AddingLocal*. El cambio de estado en la entidad remota se verá estimulado por la recepción de un mensaje del tipo DSA-REQ, provocando que el DSF en la entidad remota pase del estado *Null* al estado *Adding Remote* (ver Fig. 2).

IV. MODELADO Y SIMULACIÓN

A continuación, se modela y simula el comportamiento deseado del protocolo MAC para la gestión de los DSF exhibido en las Fig. 1 y 2. Para ello, se definieron tres niveles de jerarquía. Las actividades de modelado y simulación fueron llevadas a cabo utilizando para ello CPN Tools [20].

La conformación jerárquica de los tres niveles se puede

observar en la Fig. 3. El nivel más alto de la jerarquía etiquetado como **DSFSSTopLevel** se corresponde con el comportamiento esperado del protocolo de la Fig. 1.

El segundo nivel etiquetado como **DSFSTT**, modela el comportamiento esperado del protocolo desde el nivel más alto de abstracción representado en la Fig. 2.

El tercer nivel representado por los módulos **DSFAdd**, **DSFChg** y **DSFDelete** modela el comportamiento esperado del protocolo a nivel general de las transacciones. El módulo **DSFAdd** modela los eventos que ocurren en los sub-estados: *AddingLocal*, *AddFailed*, *AddingRemote*; el módulo **DSFChg** modela los eventos que ocurren en los sub-estados: *ChangingLocal* y *ChangingRemote*; finalmente, el módulo **DSFDelete** modela los eventos que ocurren en los sub-estados: *Deleting* y *Deleted*. En los sub-módulos **DSFAdd**, **DSFChg** también se modela las transiciones de estado al estado *Nominal*.

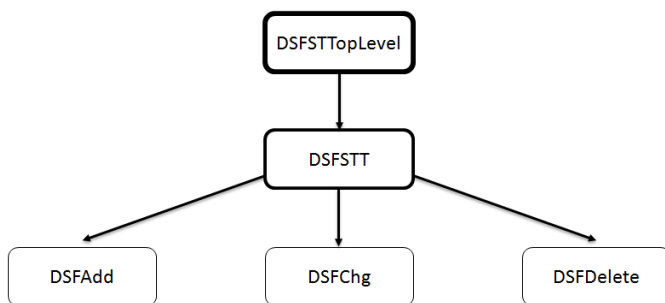


Fig. 3. Estructura Jerárquica de los Módulos del Modelo para el Protocolo MAC para la Gestión de los DSF.

En la Fig. 4 (a), se presenta el módulo de nivel superior **DSFSSTopLevel** (que se corresponde con el nivel más alto de de la Fig.3) modelado en CPN Tools. Así mismo, la Fig. 4(b), muestra el módulo **DSFSTT**, y la Fig. 5 modela el módulo asociado a la transacción **DSFAdd**. Por razones de espacio, en este trabajo sólo se describirá el módulo mostrado en la Fig.5.

Las elipses en los módulos del modelo (denominadas plazas en las CPN) descritos con la inscripción “DSFGlobalState”, permiten modelar el estado global, y los sub-estados de los DSF y de las transacciones. El estado global puede ser *NULL* o *Nominal*. Y los posibles sub-estados pueden ser: *AddingLocal*, *AddFailed*, *AddingRemote*, *ChangingLocal*, *ChangingRemote*, *Deleting*, *Deleted*, *Null* o *Nominal*. Estos son representados como una dupla (estado global, sub-estado).

En el modelo se pueden distinguir dos tipos de elementos en forma de rectángulos. Aquellos con doble borde, denominados “Transiciones de Substitución” (Fig. 4), que permiten enlazar los diferentes módulos de la jerarquía del modelo. Por su parte, los rectángulos con borde simple denominados “Transiciones”, permiten modelar las acciones que representan la ocurrencia de un evento, por ejemplo, la acción que dispara la transición de un estado a otro (ver Fig. 5).

Las transiciones y las plazas se conectan a través de arcos dirigidos de entrada, entrada/salida, salida. Para que se dispare una transición, es necesario que la plaza de entrada a una transición posea una marca (*token*) con los valores indicados en el arco de entrada a la misma. La ocurrencia de una transición podría ocasionar el cambio de estado de la plaza.

Algunas transiciones poseen una inscripción en la esquina superior izquierda, denominada guarda [5], [6]. Ella permite definir condiciones que deben cumplirse para que ocurra la transición. Por ejemplo, en la Fig. 5 la guarda [SSAdd=AddingRemote orelse SSAdd=AddingLocal] especifica que para que ocurra la transición *DSAFailed*, el valor contenido en la variable **SSAdd**, puede obtener dos diferentes valores dentro de la dupla (NULL, SSAdd). Tales valores pueden ser “AddingRemote” o “AddingLocal”.

El análisis de cada nivel de abstracción del protocolo ha sido ejecutado desarrollando la simulación de cada uno utilizando las herramientas presentes en CPN Tools, las cuales proporcionan un importante conjunto de facilidades para llevar a cabo estas tareas, y han sido ampliamente utilizadas en otras investigaciones para realizar la validación funcional de protocolos de comunicación [5], [6], [7], [9], [10], [11], [12].

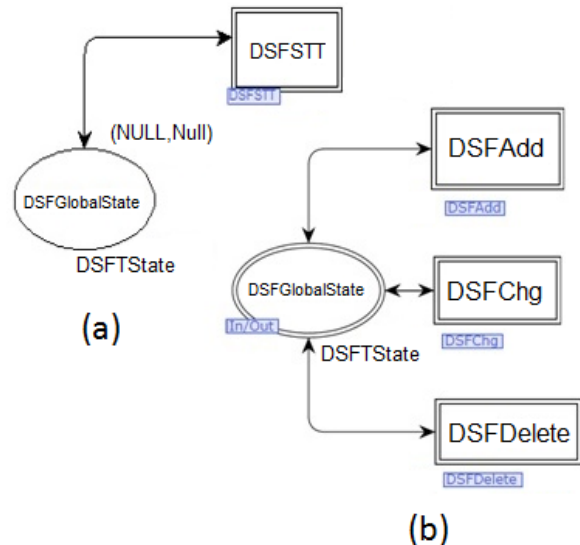


Fig. 4. Módulos de Modelo CPN 6 (a) nivel 0: DSFSSTopLevel y 6 (b) nivel 1: DSFSTT.

V. ANÁLISIS

A. Análisis Funcional Utilizando el Método del Grafo de Ocurrencias (OG)

El modelo ha sido analizado utilizando la herramienta de simulación interactiva, y el método del OG, y su correspondiente grafo (SCC, Strongly Connected Component), proporcionado por las CPNs y CPN Tools. El OG es un grafo dirigido que incluye todos los posibles estados que pueden ser alcanzados desde el estado inicial. Para efectos del modelo desarrollado, el estado inicial es (NULL,Null), dupla que representa el estado de la transacción “NULL” y el estado del DSF. En la Fig. 6, se presenta el OG resultante para el modelo del protocolo MAC para la gestión de los DSF.

En la figura es posible observar que el OG generado es manejable, pues posee un pequeño número de nodos (11) y arcos (42), lo cual facilita el análisis del mismo.

Es posible observar en la Fig. 6 que siempre que no se haya establecido un DSF, la entidad iniciadora podrá solicitar el establecimiento de un DSF tantas veces como se requiera. Esto es posible observarlo en el ciclo que se forma entre los nodos 1, 3, y 4. Es decir, estando en el estado de (NULL,

Null) se solicita el establecimiento de un DSF (por ejemplo, a través de la recepción de un mensaje del tipo DSAREQ), e inmediatamente es rechazada por la MAC iniciadora tal solicitud (por ejemplo, a través de la ocurrencia de un evento del tipo *DSAFailed*), por lo cual se retorna al estado (NULL,

Null). Así mismo, una vez establecido un DSF, se podrán solicitar tantos cambios o terminaciones (siempre que no sea aceptada la terminación) del mismo como se requiera. Esto ocurre en el conjunto de estados representados por los nodos 5, 7, 8, 9 y 11.

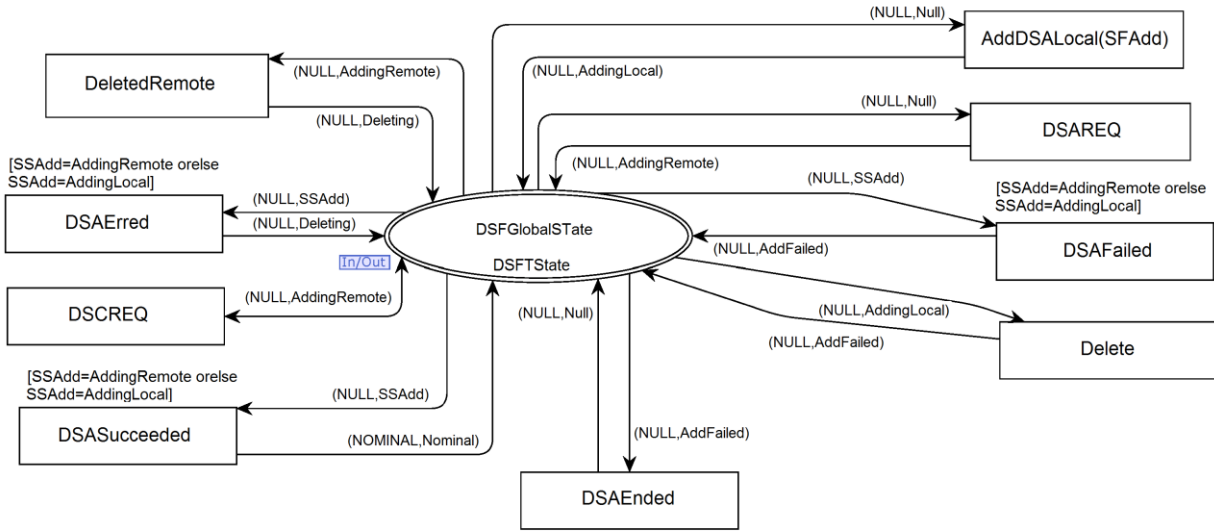


Fig. 5. Módulo CPN nivel 2: DSFAAdd.

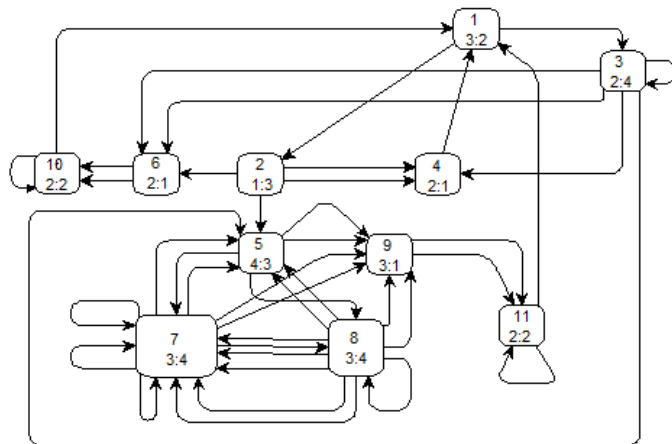


Fig. 6. OG del modelo CPN del protocolo MAC para la gestión de los DSF asociado a la Figura 2.

De los resultados arrojados por el OG en el archivo de estadísticas creado por CPN Tools, es posible observar que no existen transiciones muertas, ni marcados muertos, por lo tanto el modelo está libre de *Deadlocks* (abrazos mortales).

Una transición muerta, es aquella que no está habilitada para su ocurrencia en ningún marcado (o estado) alcanzable en el modelo [7], [8]. Un marcado muerto (estados muertos), significa que ninguna transición puede ocurrir a partir de dicho marcado (estado de la red). Este hecho está asociado a que los resultados estadísticos muestran que todas las transiciones se encuentran completamente vivas, ya que pueden ocurrir al menos una vez en una secuencia de ocurrencias para cada marcado (o estado) de la red, que es alcanzable desde el marcado inicial. Estos resultados son completamente acordes a lo esperado.

B. Análisis Aplicando Máquinas de Estado Finito

El modelo también ha sido analizado aplicando técnicas para transformar el OG asociado al modelo a un FSA que logra representar el comportamiento del sistema modelado. Para llevar a cabo tal actividad fue utilizado el algoritmo planteado por Villapol en [7], implementado e integrado a CPN Tools en [16]. Este análisis consistió en obtener los FSA minimizados, y compararlos con los diagramas descriptivos del protocolo presentados en las Fig. 1 y 2.

En la Fig. 7 se observa el FSA correspondiente al modelo del diagrama de la Fig. 1. Acá se puede resaltar que éste es el modelo ideal en un protocolo para el establecimiento de un SF. Así mismo, en la Fig. 8 se presenta el FSA correspondiente al modelo del diagrama de la Fig. 2.

El FSA de la Fig. 7 está compuesto por dos estados, los cuales a su vez son estados terminales (s0 y s1). En este contexto es definido un estado terminal, como aquel que determina la finalización de una secuencia de transacciones (caso de la Fig. 7) o la finalización de un procedimiento de gestión de un DSF (caso de la Fig. 8).

El FSA de la Fig. 8 está compuesto por tres estados, de los cuales dos de ellos son terminales (s2 y s1). En la Fig. 7, el estado “s0” representa tanto un estado inicial, como un estado terminal. En el caso de las secuencias de ocurrencia de las transacciones de la Fig. 7, éstas inician desde el estado de *NULL*. Para el caso de las operaciones involucradas en la gestión de los DSF del FSA de la Fig. 8, un DSF inicia en el estado (NULL,null), tanto para el estado de la transacción, como para el estado del DSF. Los estados terminales para la máquina de la Fig. 7 son: *NULL* (s0) y *Operational* (s1). Por su parte, los estados terminales para el FSA de la Fig. 8 son: (NULL,null) estado s1 o (NOMINAL, Nominal) estado s2.

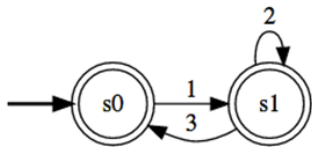


Fig. 7. FSA correspondiente al modelo CPN de la Fig. 1.

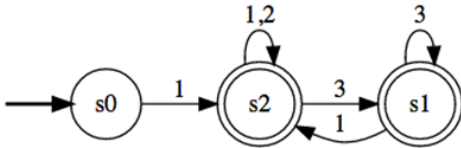


Fig. 8. FSA correspondiente al modelo CPN de la Fig. 2.

Las inscripciones en los arcos de las Fig. 7 y 8, tienen la nomenclatura 1, 2, y 3. Tales identificadores representan las transiciones en los modelos CPN desarrollados.

En la Tabla 1, se presenta el detalle de las transiciones del modelo CPN desarrollado para el diagrama de la Fig. 1, quienes representan las ocurrencias de eventos que disparan cambios de estado (están relacionadas a los tipos de transacciones que pueden realizarse sobre un flujo de servicio), y sus correspondientes identificadores asociados.

TABLA I
IDENTIFICADORES PARA LAS TRANSICIONES DE ESTADO DEL OG MINIMIZADO DE LA FIGURA 3

Modelo	Transacción (Transición)	ID
Transiciones de Estado de las	DSA	1
Transacciones para la gestión de Flujos de Servicios Dinámicos (Fig. 3)	DSC	2
	DSD	3

En la Tabla 2, se muestran también los identificadores asignados a las transiciones en el modelo CPN desarrollado para el diagrama de la Fig. 2, quienes representan las ocurrencias de eventos que disparan cambios de estado en los DSF, debido a la ocurrencia de las transacciones asociadas a éstos.

Todas las transiciones con ID=1, representan las transiciones asociadas a la transacción del tipo DSA. Las transiciones con ID=2, representan las transiciones asociadas a la transacción del tipo DSC. Finalmente, las transiciones con ID=3, representan las transiciones asociadas a la transacción del tipo DSD.

Al comparar los FSA minimizados de las Fig. 7 y Fig. 8, es posible observar que los diagramas de transición de estados para las transacciones, y los DSF de las Fig. 1 y Fig. 2 de la especificación del estándar no son equivalentes.

A fin de validar este resultado, se procedió a generar y comparar los lenguajes correspondientes a tales FSA (Fig. 7 y Fig. 8). En este contexto, un alfabeto según Hopcroft, Motwani and Ullman [21], es un conjunto finito no vacío de símbolos. Se denota generalmente como Σ . Siguiendo esta definición, el alfabeto para las FSA de las Figuras 7 y 8, es $\Sigma=\{1,2,3\}$.

Así, dado un alfabeto Σ , una cadena sobre un alfabeto (o palabra) es una secuencia finita de símbolos de dicho alfabeto [21], [22]. El conjunto Σ^* incluye todas las cadenas posibles que se pueden formar a partir de los símbolos de un alfabeto

dado. Por ejemplo, si $\Sigma=\{1,2,3\}$, entonces las posibles cadenas derivadas de este alfabeto pueden ser 1, 2, 3, 12, 123, 13, 23,..., etc. Para cualquier alfabeto, puede existir la cadena vacía (o épsilon) representada por el carácter ϵ [21], [22].

TABLA II
IDENTIFICADORES DE LAS TRANSICIONES DE ESTADO DEL OG MINIMIZADO DEL MODELO DESARROLLADO PARA LA FIG.4

Modelo	Módulos en el modelo	Transiciones en los módulos	ID
Transiciones de Estados de los Flujos de Servicio Dinámicos para el modelo CPN de Fig. 4.	DSFAdd	AddDSALocalSFAdd	1
		DSAEnded	3
		DSAErred	1
		DSAFailed	1
		DSAREQ	1
		DSASucceeded	1
		DSCREQ	2
		Delete	3
		DeletedRemote	3
		DSFChg	Change
	ChangeRemote		2
	DSAACKLost		1
	RemoteDSAACKLost		1
	DSCACKLost		2
	DSCErred		2
	DSCFailed		2
	DSCReq2		2
	DSCREQBOnly		2
	DSCSucceeded		2
		DSCREQ2	2
DSCREQSSOnlySFChangeRemote		2	
Delete			
NewDSCREQ		3	
SFChangeBOnly		2	
DSFDeleted		DSDEndedyNumXacts0	3
	DSDErred	3	
	DSDREQRemote	3	
	DSDSucceeded	3	
	DSD_Succeeded	3	
	DeleteLocal	3	

Por lo tanto, el lenguaje L reconocido por un autómata dado, es un subconjunto de cadenas (o símbolos) formadas a partir de un alfabeto Σ , luego $L \subset \Sigma^*$. L contiene todas las posibles cadenas que se puedan formar a partir de los símbolos de dicho alfabeto siguiendo las transiciones del FSA [21].

Si todas las cadenas de un lenguaje satisfacen alguna regla en común, el lenguaje puede ser definido mediante una expresión regular.

La Tabla 3 muestra las expresiones regulares asociadas al lenguaje correspondiente al FSA de la Fig. 7, a la que se denominará L_T (Lenguaje de las Transacciones). La notación $X \rightarrow Y$ indica el subconjunto de cadenas generadas, cuyo procesamiento comienza en el estado X del autómata, y finaliza en el estado Y . El estado X siempre corresponderá al nodo inicial del autómata (estado inicial), mientras que el estado Y corresponderá a cada uno de los estados terminales.

Ejemplos de cadenas aceptadas por el FSA de la Fig. 7, según las expresiones regulares definidas en la Tabla 3 son:

- $0 \rightarrow 0$: DSA, DSD;
DSA, DSC, DSD;
DSA, DSC, DSC, DSD
- $0 \rightarrow 1$: DSA;
DSA, DSC;
DSA, DSD, DSA;
DSA, DSC, DSD, DSA

TABLA III

EXPRESIONES REGULARES ASOCIADAS AL LENGUAJE CORRESPONDIENTE AL FSA DE LAS TRANSACCIONES (L_T), PRESENTADA EN LA FIG. 7

Estado Inicial → Estado Terminal	Expresión Regular que de denota el Lenguaje
0 → 0	(12*3)*
0 → 1	(12*3)*12*

Así mismo, la Tabla 4 muestra las expresiones regulares asociadas al lenguaje correspondiente al FSA de la Fig. 8, a la que se denominará L_{DSF} (Lenguaje de los Flujos de Servicios Dinámicos).

TABLA IV

EXPRESIONES REGULARES ASOCIADAS AL LENGUAJE CORRESPONDIENTE A LA MÁQUINA DE ESTADO DE LOS DSF (L_{DSF}), PRESENTADA EN LA FIG. 8

Estado Inicial → Estado Terminal	Expresión Regular que de denota el Lenguaje
0 → 2	1(1 2 33*1)*
0 → 1	1(1 2 33*1)*33*

Ejemplos de cadenas aceptadas por el FSA de la Fig. 9 según las expresiones regulares definidas en la Tabla 4 son:

- 0 → 2: DSAREQ, DSASucceeded;
AddDSALocalSFAdd, DSASucceeded;
DSAREQ, DSASucceeded, DSCReq2, DSCSucceeded;
AddDSALocalSFAdd, DSASucceeded, Change, DSCSucceeded;
DSAREQ, DSCREQ, DeletedRemote, DSDucceeded, DSDEndedyNumXacts0;
AddDSALocalSFAdd, Delete, DSAEnded;
AddDSALocalSFAdd, DSASucceeded, DSCReq2, RemoteDSCACKLost, DSCSucceeded;
- 0 → 1: DSAREQ, DSCREQ, DeletedRemote, DSDSucceeded, DSDEndedyNumXacts0, AddDSALocalSFAdd, Delete, DSAEnded;
AddDSALocalSFAdd, DSAFailed, DSAEnded;
DSAREQ, DSAFailed, DSAEnded;
DSAREQ, DSASucceeded, DSCReq2, DSCErred, DSDSucceeded, DSDEndedyNumXacts0;
DSAREQ, DSASucceeded, Change, DSCREQBSONly, DSAACKLost, DSCACKLost, Delete, DSDSucceeded, DSDEndedyNumXacts0;

C. Comparación de Lenguajes entre el FSA de las Transacciones y el FSA para la Gestión DSF

A fin de realizar un análisis más completo sobre las máquinas de estado de las Fig. 7 y Fig. 8, se procedió a realizar una comparación entre los lenguajes generados por ambas máquinas (L_T vs L_{DSF}), utilizando para ello la función *Difference* proporcionada por Open FST [23] e implementada en [13].

La comparación genera una máquina que es capaz de generar todas las cadenas que acepta el autómata de la Fig. 7 (lenguaje L_T), y que no son aceptadas por el autómata de la Fig. 8 (lenguaje L_{DSF}), es decir ($L_T - L_{DSF}$). Así mismo, se procedió a realizar el proceso inverso ($L_{DSF} - L_T$), es decir, el resultado es una máquina con todas las cadenas aceptadas por el autómata de la Fig. 8, y que no son aceptadas por el autómata de la Fig. 7.

El lenguaje resultante (L_r) de la máquina resultante $L_T - L_{DSF}$ se define de la siguiente manera:

$$L_r = L_T - L_{DSF} = \{p \mid p \in L_T \text{ and } p \notin L_{DSF}\} \quad (1)$$

El lenguaje resultante (L_r) de la máquina resultante $L_{DSF} - L_T$ se define de la siguiente manera:

$$L_r = L_{DSF} - L_T = \{p \mid p \in L_{DSF} \text{ and } p \notin L_T\} \quad (2)$$

El resultado de aplicar (1) generó una máquina de estado vacía, puesto que L_T está contenido en L_{DSF} . Esto se debe a que el FSA de la Fig. 7 es capaz de generar todas las cadenas o secuencias de símbolos que son posibles obtener del FSA de la Fig. 8.

El resultado de aplicar (2) generó una máquina no vacía, lo que indica que existen diferencias entre $L_{DSF} - L_T$. Es decir, existen cadenas aceptadas por el FSA de la Fig. 7, que no son aceptadas por el FSA de la Fig. 8. En la Fig. 9, se presenta la máquina resultante de aplicar (2).

La Tabla 5 muestra las expresiones regulares asociadas al lenguaje correspondiente al FSA de la Fig. 9.

Ejemplos de cadenas aceptadas por el FSA de la Fig. 9, según las expresiones regulares definidas en la Tabla 5 para L_r son:

- 0 → 2: AddDSALocalSFAdd, Delete, DSAEnded, AddDSALocalSFAdd;
AddDSALocalSFAdd, DSASucceeded;
DSAREQ, DSASucceeded;
AddDSALocalSFAdd, DSASucceeded;
DSAREQ, DSCREQ, DSASucceeded;
AddDSALocalSFAdd, DSASucceeded, DSCReq2, DSCFailed;
DSAREQ, DSASucceeded, Change, DSCSucceeded;
AddDSALocalSFAdd, DSASucceeded, Change, DSCFailed;
- 0 → 4: AddDSALocalSFAdd, Delete, DSAEnded;
AddDSALocalSFAdd, DSAFailed, DSAEnded;
DSAREQ, DSAFailed, DSAEnded;
DSAREQ, DSCREQ, DSASucceeded, DSCREQ, DSCErred, DSDSucceeded;
AddDSALocalSFAdd, DSDREQRemote, AddDSALocalSFAdd, DSASucceeded, Change, Delete, DSDSucceeded;
DSAREQ, DSASucceeded, DSCReq2, NewDSCREQ, DSCErred, DSDErred, DSDEndedyNumXacts0;

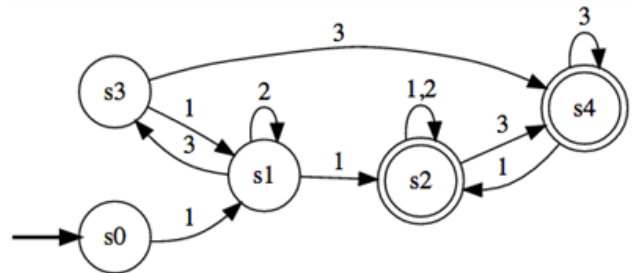
Fig. 9. Máquina resultante de calcular L_r .

TABLA V

EXPRESIONES REGULARES ASOCIADAS AL LENGUAJE CORRESPONDIENTE AL FSA L_r DE LA FIG. 9.

Estado Inicial → Estado Terminal	Expresión Regular que de denota el Lenguaje
0 → 2	1(2 31)*(1 333*1)(1 2 33*1)*(1 2 ε 33*1)
0 → 1	1(2 31)*(3 (1 333*1)(1 2 33*1)*33*

Este resultado evidencia las diferencias, ambigüedades y vacíos de información presentes en los diagramas de transición de estados de los DSF y las transacciones que los soportan, en las especificaciones del estándar IEEE 802.16 [3], [4], [17], [18], [19]. Tales inconvenientes observados en los documentos de la especificación podrían conllevar a graves errores de implementación de parte de los fabricantes de la tecnología, debido a la interpretación individual al momento

de las implementaciones. Estos errores de implementación podrían ocasionar, por ejemplo, que dos entidades intentando mantener establecidas conexiones para el transporte de datos, no logren establecer de forma completa y satisfactoria las conexiones correspondientes a los SF, o que las mismas no sean eliminadas o liberadas satisfactoriamente, conllevando a la retención de recursos (conexiones) no utilizados.

VI. CONCLUSIONES

La reservación de recursos en redes con características de tráfico variable tales como las redes móviles e inalámbricas no es una tarea fácil. Estos procedimientos podrían llegar a consumir mucho tiempo, y cierta información desactualizada podría permanecer almacenada en los dispositivos que conforman la red, incluso si ya no forma parte de los flujos de datos.

Un protocolo que permite la gestión de reservación de recursos en forma de flujos de servicios en redes móviles e inalámbricas de banda ancha es el protocolo MAC WiMax para la gestión de DSF. El protocolo utiliza un enfoque soft-state para lidiar con las características variables de los SF, lo cual resulta altamente deseable en estos entornos. Sin embargo, el protocolo resulta sumamente complejo de entender. El protocolo ha sido analizado en este trabajo a fin de mostrar las imprecisiones y debilidades presentes en la especificación (como las descritas al finalizar la sección III), y cómo estos aspectos pueden afectar las implementaciones del protocolo. También hemos mostrado el potencial que representa la utilización del enfoque orientado a transacciones en la descripción de este tipo de protocolos.

La actividad de modelado, llevó a determinar un conjunto de ambigüedades y vacíos de información a nivel de los cambios de estado de los DSF, y cómo éstos se relacionan con los cambios de estados en las transacciones que los soportan, que no están claros en la especificación.

El resultado de aplicar el análisis de los modelos desde la perspectiva del OG, y aplicando técnicas de conversión y reducción a un FSA, permitió validar la funcionalidad del Protocolo WiMax MAC para la gestión de los DSF. En la etapa de análisis se logró la determinación de expresiones regulares asociadas al lenguaje de los FSA de ambos modelos, así como también presentar y analizar conjuntos de cadenas asociadas al lenguaje. La comparación de los lenguajes, permitió evidenciar la confusa relación que existe entre el diagrama de estado de las transacciones que soportan la gestión de los DSF y el diagrama de estado de los DSF. Esto se evidencia en el resultado obtenido en el FSA $L_r \cdot (L_{DSF} - L_T)$.

Debido a que el núcleo del estándar IEEE 802.16 es la base fundamental para generaciones de redes inalámbricas de alta velocidad y cobertura amplia, como las redes LTE (*Long Term Evolution*) y las redes 5G, es preciso que a través de actividades como la validación y verificación de protocolos sea aplicada, como en el presente trabajo.

Resulta deseable que se tomen en cuenta tales observaciones en las especificaciones del estándar IEEE 802.16, con el objeto de que el núcleo de ésta tecnología sea utilizado en la especificación de las redes 5G para soportar sus requerimientos de QoS, las cuales están enfocadas en soportar la creciente y sostenida demanda de aplicaciones multimedia.

REFERENCIAS

- [1] I. Bernal, "A Friendly Introduction to the Requirements and Supporting Technologies for 5G Cellular Networks," *Revista Politécnica*, vol. 37, no. 1, Mar. 2016.
- [2] IEEE Sta. 802.16-2001, "Local and Metropolitan Area Network, Part 16: Air Interface for Fixed Broadband Wireless Access Systems," Oct. 2002.
- [3] IEEE Sta. 802.16-2004. "Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems" Oct. 2004.
- [4] L. M. Kristensen and K. I. Fagerland, "Applications of Coloured Petri Nets for Functional Validation of Protocol Designs," *Trans. On Petri Nets and Other Models of Concurrency. Lecture Notes in Computer Science*, vol. 7480, pp 56-115, 2013.
- [5] K. Jensen and L. Kristensen, "Coloured Petri Nets: Modelling and Validation of Concurrent Systems," Springer-Verlag, Berlin, 2009.
- [6] M.E. Villapol, "Modelling and Analysis of the Resource Reservation Protocol Using Coloured Petri Nets," PhD Thesis, University of South Australia, Australian, Nov. 2003.
- [7] J. Billington, "Formal Specification of Protocols: Protocol Engineering," *Encyclopedia of Microcomputers*, Marcel Dekker, New York, Vol. 7, pp 299-314, 1991.
- [8] A. Morales, M. Villapol and L. Contreras, "Improving the Specification of the Wimax ARQ Operation," *Proc. XL Conferencia Latinoamericana en Informática (CLEI 2014)*. ISBN: 978-1-4799-6129-0, DOI:10.1109/CLEI.2014.6965138, IEEE Xplore, INSPEC Accession Number: 14760705, Sep. 2014.
- [9] A. Morales and M.E. Villapol, "A Formal Review of Service Specification for the Management of the Connections in the IEEE 802.16 MAC Layer," *CLEI Electronic Journal*, vol. 16, no. 2, paper 2, ISSN 0717- 5000, Aug. 2013.
- [10] A. Morales and M.E. Villapol, "Towards Formal Specification of the Service in the IEEE 802.16 MAC Layer for Connection Management," *Proc. 9th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS '10)*, ISSN: 1792-6998, ISBN: 978-960-474-257-8, pp. 140-146. Dec. 2010.
- [11] A. Morales y M. Villapol, "Modelado y Análisis Inicial de la Especificación de Servicio de la capa MAC del IEEE 802.16 utilizando Redes de Petri Coloreadas (CPN)," *XXXIII Conferencia Latinoamericana en Informática (CLEI2007)*, ISBN: 978-9968-9678-9-1, San José, Costa Rica, Octubre 2007.
- [12] H. Guesmil, S. Maaloul, R. Tourki, "Design of Scheduling Algorithm for QoS Management on WiMAX Networks," *Computer Science and Engineering Journal*, Vol. 2, pp 43-50, DOI: 10.5923/j.computer.20110102.08, 2011.
- [13] J. Kuo, C. Shih, y Y. Chen, "A Host-based Fast Mobility Scheme (HFMS) in 802.16j Mobile RS Mode," *Journal of Advances in Computer Network*, Vol. 1, No. 3, Sep. 2013.
- [14] M. Ahamed, W. Fernando, P. Permuna, K. Jayathilake, S. Mathurusha, N. Vithanage y D. Dhammearatchi, "Enhanced QoS support in OFDMA-Based WiMAX Systems," *COMPUSOFT, International Journal of Advanced Computer Technology*, vol. 5(4), Ap. 2016.
- [15] J. Carrasquel, "Java/PROSEGA: extensión de CPN Tools para la generación de Lenguajes de Automatas y la reducción de Grafos de Estado a Máquinas de Estado Finito," Degree thesis, Faculty of Sciences, School of Computing, Central University of Venezuela. Tutors: Ana Morales and María Villapol. Aug. 2015.
- [16] IEEE Sta. 802.16e-2005, "Local and Metropolitan Area Network, Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment 1: Management Information Base," Dec. 2005.
- [17] IEEE Sta. 802.16g-2007, "IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 3: Management Plane Procedures and Services," Dec. 2007.
- [18] IEEE Sta. 802.16-2009, "Local and Metropolitan Area Network, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems," May. 2009.
- [19] CPN Tools. <http://cpntools.org/start>. 2017.
- [20] J. Hopcroft, R. Motwani and J. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison-Wesley Publishing Company, second edicion, Madrid, ISBN: 84-7829-056-7, 2002.
- [21] M. Sipser, "Introduction to the Theory of Computation," PWS Publishing Company, Boston, Massachusetts, ISBN 0-534-94728-X, 1997.
- [22] OpenFST Library. Download Section. <http://www.openfst.org/twiki/bin/view/FST/FstDownload>. Feb. 2016.