

Sincronización de la información de tasación en entornos convergentes

Charging information synchronization in convergent environment

Mary Cristina Carrascal-Reyes^{1a}, José Luis Arciniegas-Herrera^{1b}

¹Grupo de Ingeniería Telemática, Universidad del Cauca, Colombia. Orcid: ^a0000-0002-8962-7293, ^b0000-0002-1310-9123.
Correos electrónicos: ^amcarrasc@unicauca.edu.co, ^bjlarci@unicauca.edu.

Recibido: 24 octubre, 2017. Aceptado: 2 mayo, 2018. Versión final: 18 junio, 2018.

Resumen

Los servicios y los sistemas de telecomunicaciones se encuentran actualmente en un proceso de transformación. Los usuarios son cada vez más diversos en sus necesidades y requerimientos, que demandan más sofisticación y servicios de comunicación con características individualizadas que mejoren su experiencia. En este contexto, la gestión de la información toma un papel preponderante, y la información de tasación es un elemento clave en la provisión de servicios en un entorno de convergencia.

Actualmente, los datos de tasación se encuentran distribuidos entre los diferentes elementos de red envueltos en el despliegue de un servicio, lo que implica contar con un mecanismo que mantenga la consistencia de la información en la prestación de los servicios convergentes. Este artículo propone un mecanismo para asegurar la sincronización y la consistencia de datos en un entorno IMS (IP Multimedia Subsystem), mediante técnicas de replicación de bases de datos ya evaluadas en el campo de las tecnologías de la información.

Palabras clave: IP Multimedia Subsystem; *middleware*; perfil de usuario; sincronización de bases de datos; tasación.

Abstract

The services and systems are in the midst of a major transformation. Users are becoming more diverse in their needs and requirements, resulting in an increased demand for sophisticated, individualized communications services that improve their experience. In this context, information management is an important aspect, so charging information is a key element when we want to provide services in a converged environment.

Currently, charging data are distributed between different network elements involved in service provisioning. Therefore, it is a challenge to implement a mechanism to keep data consistency that allows for a real services convergence.

This article proposes a mechanism to ensure data synchronization and consistency in IMS environments. To achieve this goal, we adopted databases synchronization techniques already tested in the information technology field.

Keywords: charging; group communications; IP Multimedia Subsystem; *middleware*; synchronization; user profile.

1. Introducción

Actualmente los operadores de telecomunicaciones se encuentran ante grandes desafíos: la evolución es una obligación, la prestación de servicios que se ajusten a las

necesidades de los clientes y que los posicionen un paso delante de sus competidores es apremiante. Sin embargo, posicionar servicios diferenciados y proporcionar una experiencia superior al cliente son hechos que causan un profundo impacto en los sistemas subyacentes de AAA

(*Authentication, Authorization, Accounting* - Autenticación, Autorización, Contabilidad) e implican transformaciones y cambios en los procesos establecidos. En esta inevitable evolución, los servicios de gestión juegan un rol fundamental. Poder ofrecer alternativas personalizadas para el cobro de los servicios es clave en la retención o no de los clientes, pero también puede ser una herramienta para lograr un mejor uso de los recursos de los que se dispone [1][2].

Dadas todas las nuevas exigencias del mercado, la capacidad en tiempo real es fundamental para las nuevas ofertas de servicios de telecomunicaciones que requieren el uso de sistemas de recarga en línea y de políticas de gestión, pero también es importante para otros tipos de nuevos servicios (como la facturación directa al operador, por ejemplo). El paso a un entorno en tiempo real para mejorar la experiencia del cliente permite a las plataformas de servicios de comunicaciones presentar ofertas que son más relevantes para los clientes actuales y potenciales, y, por lo tanto, ayuda a generar nuevas fuentes de ingresos. También proporciona una mejor y más completa información del comportamiento de los clientes y permite ofrecer opciones de autoservicio [2][3]. Cada una de estas características ayudaría a reducir la rotación de los usuarios y aumentaría la probabilidad de que los clientes actuales recomienden los servicios a otros [4][5].

Para poder prestar servicios de calidad en las redes convergentes, es necesario tener una colección de datos relacionados con el usuario y el uso de los servicios, que esté actualizada consistentemente y que pueda ser accedida con rapidez, en cualquier momento, desde cualquier lugar y sin importar la carga que pueda tener la red [17].

El 3GPP (3rd Generation Partnership Project) ha definido un mecanismo de sincronización en algunas interfaces específicas, la Sh y Cx de IMS; sin embargo, no se ha tratado de mejorar el desempeño de los nodos comprometidos mientras se garantiza la consistencia de la información. Y, por otra parte, tampoco existe un mecanismo genérico para integrar cualquier módulo de IMS o de otra red que necesite información de usuario para prestar o tasar sus servicios.

Así, se observa la necesidad de definir un mecanismo de sincronización de información que sea fácil de implementar y capaz de interoperar con las distintas bases de datos que se encuentran en una red IMS y en general en las NGN (Next Generation Network, Redes de Próxima Generación). Por ejemplo, en un entorno convergente en el que un usuario esté usando el servicio de IPTV (Internet Protocol Television) mientras hace uso del transporte público, es deseable, desde el punto de

vista del operador, conocer las características del contenido al que está teniendo acceso, como la clase, la calidad, el tipo de dispositivo que está siendo usado, el ancho de banda utilizado, la ubicación precisa en el momento de uso del contenido, los servicios adicionales que ha usado, etc., para así poder realizar procesos de cobro y adecuar dinámicamente opciones personalizadas en la prestación de los servicios. Se podría, de acuerdo a su suscripción, inyectar publicidad de productos o locales comerciales que se ubiquen en la zona geográfica en la que se encuentra, así como realizar otros análisis más avanzados. Además, sería posible analizar los datos de los contactos del usuario en las redes sociales, para sugerir un encuentro en una cafetería de la zona, o informar acerca de los contactos que ya vieron o están viendo el contenido al cual está teniendo acceso en ese momento y así poder intercambiar, por ejemplo, comentarios. Lograr este tipo de dinámica en la prestación de servicios de telecomunicaciones impone importantes retos, puesto que actualmente la información del usuario y su comportamiento en el uso de los servicios son datos que se encuentran distribuidos en diferentes elementos de red: en el núcleo se contará con la información sobre ancho de banda, tipo de dispositivo, servicios habilitados, tipo de terminal, hora de acceso, etc., pero los detalles acerca del servicio IPTV se encontrarán en el servidor de aplicaciones que presta el servicio (tipo de contenido, formato, etc.), y las coordenadas de la ubicación del usuario estarán dadas por un AS (Application Server, Servidor de Aplicaciones) que cumple esta tarea.

En este artículo se exponen los conceptos implicados en la construcción de la solución propuesta. En la primera parte se describen los principios de funcionamiento de IMS, en la segunda parte se describen los conceptos asociados a la tasación, así como la descripción de los sistemas de tasación que se encuentran actualmente estandarizados, con el fin de comprender los vacíos de las implementaciones actuales y así plantear el alcance del diseño que se propone en este trabajo. En el ítem cinco se describe en qué consiste la sincronización de bases de datos y en las partes seis y siete se describe la propuesta, el diseño, la implementación y los resultados de las pruebas realizadas.

2. Arquitectura IMS

Las especificaciones de IMS definen una arquitectura completa de la capa de control (por encima del dominio de conmutación de paquetes) y cubren también todos los elementos necesarios para soportar el establecimiento de sesiones multimedia en la red de paquetes. Mientras IETF (Internet Engineering Task Force, Grupo de Trabajo en Ingeniería de Internet) ha estandarizado el protocolo SIP (Session Initiation Protocol, Protocolo de

Iniciación de Sesión), el más apropiado en relación con las soluciones cliente/servidor, pero sin asociarlo con las arquitecturas, 3GPP ha definido con precisión la arquitectura y los procedimientos para hacer la capa de control y los núcleos de la red seguros, fiables y con funciones de gestión [6].

Los componentes principales relacionados con los servicios de comunicación son (figura 1) los siguientes:

2.1. Proxy Call Session Control Function (P-CSCF)

Esta entidad actúa como el punto de acceso al dominio SIP, desde una perspectiva de control de sesión [6][9].

El P-CSCF genera el ICID (IMS Charging Identifier, Identificador de Cobro IMS) utilizado en los procedimientos de cobro en los que se correlacionan las sesiones y los gastos, esto dado que el P-CSCF, como ya se expuso, es el punto de entrada en IMS.

2.2. Interrogating Call Session Control Function (I-CSCF)

Mientras el P-CSCF es el punto de acceso a la red IMS, el I-CSCF sirve como pasarela o puerta de enlace dentro de cada red IMS. El I-CSCF ayuda a proteger al S-CSCF y al HSS de accesos no autorizados por parte de otras redes. Cuando el S-CSCF está enviando peticiones o respuestas a otras redes, el mensaje primero es enviado hacia el I-CSCF, el cual a su vez lo envía hacia la red de destino [7][8].

2.3. Serving Call Session Control Function (S-CSCF)

En IMS el S-CSCF cumple el papel de núcleo principal de la red. Esta entidad controla todos los aspectos relacionados con los servicios de un suscriptor, manteniendo el estado de cada una de las sesiones que han sido iniciadas. El S-CSCF controla la mensajería y la distribución de contenidos. Este proporciona el estado del registro de un suscriptor a otras aplicaciones (servidores de aplicación) y mantiene el control sobre esos servicios mientras el dispositivo esté registrado. [9][10][11].

2.4. Home Subscriber Server (HSS)

El Home Subscriber Server (HSS) es la base de datos maestra que contiene la información de usuario y de suscriptor, y se encarga de apoyar a las entidades de red en el establecimiento tanto de llamadas como de sesiones [10]. El HSS almacena los datos del usuario, tales como los servicios a los que un suscriptor tiene acceso, las diversas identidades (la identidad de usuario privada, y todas las identidades de usuario públicas), las redes a las que el usuario tiene acceso para transitar (en el caso de

las redes inalámbricas) y la localización del dispositivo del suscriptor [9].

2.5. Servidores de Aplicación (AS)

Los Servidores de Aplicación proveen servicios específicos a los usuarios finales. IMS define tres tipos de servidores de aplicación:

- Servidores SIP
- Servidores OSA (Open Service Access, Acceso a Servicios Abiertos)
- Servidores CAMEL (Customise Applications for Mobile Networks Enhanced Logic, Aplicaciones a la Medida para Redes Móviles con Lógica Mejorada)

2.6. Actualización de la información de usuario entre AS y HSS

La interfaz Sh se encuentra definida para comunicar un SIP Application Server (SIP AS) con el HSS, con el objetivo de

- Descargar y actualizar datos de usuario
- Pedir y enviar notificaciones en cambios en datos de usuario

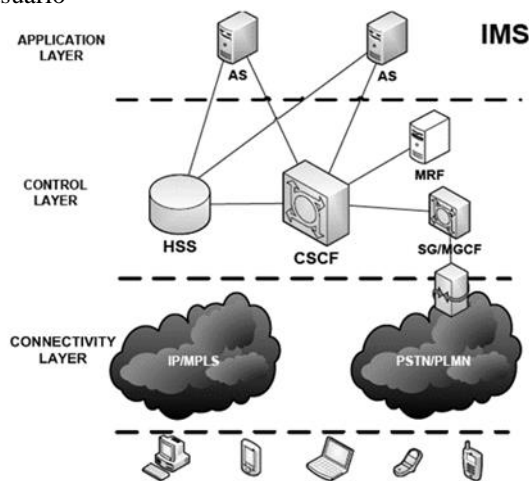


Figura 1. Arquitectura IMS. Fuente: elaboración propia.

Así, la sincronización de información de usuario entre el HSS y los AS se resume en los anteriores puntos. Para cumplir con este propósito, se usa el protocolo Diameter en su forma base, con una extensión que maneja los mensajes definidos para esta interfaz [11].

3. Conceptos asociados a la tasación e interacción de procesos

3.1. Terminología asociada

Infortunadamente, la terminología asociada a la tasación no se encuentra armonizada. Para facilitar el

entendimiento de los términos en el área, a continuación, se describen los términos y las definiciones adoptados en este artículo. Existen cuatro términos que son considerados como los más importantes relacionados con los procesos de la tasación. *Metering* se refiere al proceso de recolectar la información acerca del recurso usado en un elemento de red particular. *Accounting* usa los logs generados en el proceso de *metering* y adiciona información acerca del uso de los recursos proveniente de diferentes elementos de red. *Charging* se refiere al proceso de calcular el costo, expresándolo en unidades aceptables para la gestión de la red, de un consumo de servicio, haciendo uso de la información de *accounting*. En el proceso de *billing*, los costos son agrupados, y el procedimiento de pago del servicio es direccionado hacia la parte que consumió el servicio (ver figura 2).

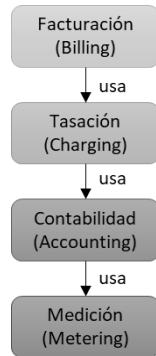


Figura 2. Procesos relacionados al *charging*. Fuente: elaboración propia.

La literatura diferencia entre los términos *charging models* y *charging mechanisms*. Un modelo de tasación define una lista de criterios que serán aplicados con el fin de calcular el costo monetario del uso de un servicio, así como una lista de precios, también conocidos como las tarifas, de una unidad de servicio definida que será usada en el cálculo de los costos, aplicando el criterio

establecido. Por ejemplo, el modelo de tasación basado en volumen usa la cantidad de datos transferida como un criterio, y define un precio por cada KB transferido. Otro modelo muy usado es el modelo basado en tiempo, el cual usa el tiempo transcurrido como criterio y el modelo basado en contenidos, el cual usa el tipo de servicio/contenido proveído como el criterio de cobro [13][14].

Un mecanismo de tasación (*charging mechanism*) puede ser ejecutado en línea (*online*) o fuera de línea (*offline*) e identifica donde se tasa el servicio, mientras este se provee, o si el proceso es pospuesto en el tiempo. Los mecanismos de tasación fuera de línea separan en el tiempo la ejecución de la tasación del despliegue del servicio. La tasación es solicitada por el servicio cuando este inicia, pero solamente los procesos de *metering* y *accounting* son iniciados. Cuando el servicio termina, se procesan los datos de *accounting*, se calcula el costo final del servicio y se envía la información al dominio del *billing*.

3.2. Procesos relacionados al *charging*

Los mecanismos de tasación en línea son ejecutados en tiempo real de acuerdo con la provisión del servicio, lo que exige a su vez que los procesos de *metering* y *accounting* también se ejecuten en tiempo real. La principal ventaja de este mecanismo es la habilidad de controlar el costo del servicio en cada punto de la sesión. Además, habilita la introducción de mecanismos de autorización de servicio (control sobre la autorización o no a los componentes de un servicio en particular) o se pueden tomar decisiones sobre la terminación en la prestación del servicio de acuerdo a ciertas condiciones y *billing*, por parte de los dos organismos de estandarización que definen estos procesos, la IETF y el 3GPP. La IETF usa el término *rating* para referirse a los que el 3GPP y este proyecto denominan *charging* [15]. El 3GPP se refiere a *accounting* como el proceso de distribuir los costos entre las partes involucradas en el

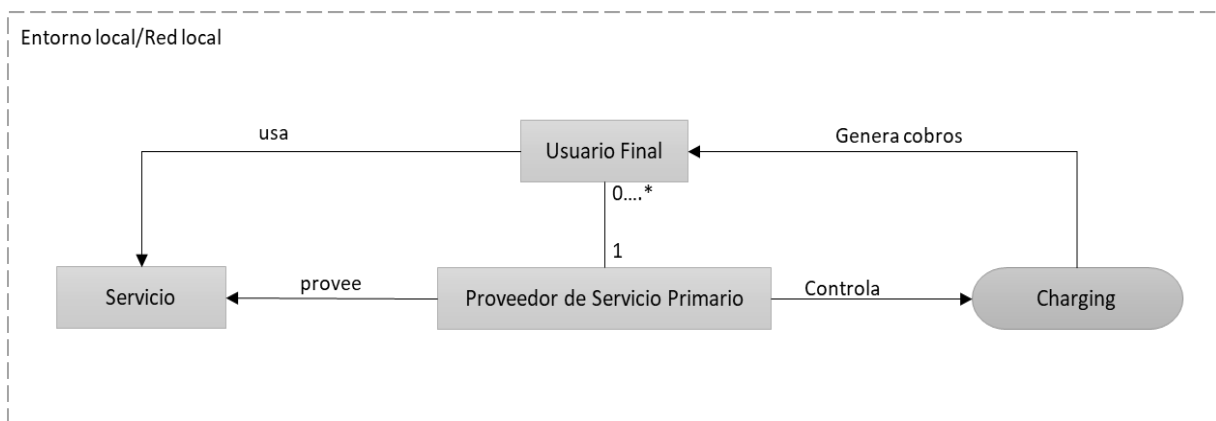


Figura 3. Descripción general del proceso de *charging*. Fuente: elaboración propia.

despliegue del servicio, y define *billing* generalmente como se ha descrito en este documento [16]. Sin embargo, en 3GPP, *charging* se usa, en sentido amplio, como un término sombrilla para todos los subprocesos involucrados. La diferencia en la terminología se explica en detalle en [17][18][19].

3.3. Actores y *stakeholders*

Se identifican dos roles en los procesos relacionados con el *charging*: usuario y proveedor. El usuario es la entidad que recibe o usa el servicio que se tasa por usarlo. Un proveedor es la entidad que provee el servicio. En general, el rol de proveedor es desempeñado por diferentes *stakeholders*, como el operador de red móvil, el proveedor de servicios de internet o un proveedor de servicio de una tercera parte. El término ‘proveedor del servicio’ será usado desde ahora para referirse a los *stakeholders*.

El rol de usuario puede ser desempeñado por cualquier persona natural que recibe un servicio o por un proveedor de servicios, esto permite que cualquier proveedor de servicios pueda prestar un servicio, pero que también pueda ser usuario de otros servicios proveídos por otros proveedores.

El PSP (Proveedor de Servicio Primario) es un proveedor de servicio que tiene una relación de negocios establecida con un usuario final, generalmente mediante un Acuerdo de Nivel de Servicio (SLA, Service Level Agreement), y es responsable de la prestación general y el control de los servicios dirigidos hacia el usuario [20]. En 3GPP, una red controlada por el PSP se denomina como Home Environment [16]. Desde el punto de vista del usuario final, su PSP es el único que es responsable por la tasación. En el caso en que un servicio sea proveído usando recursos de servicios de subproveedores adicionales, es necesario el establecimiento de procedimientos de *charging* interdominio. Diferentes aspectos de la tasación interdominio entre diferentes proveedores de servicios se discuten en [16][17][18][19].

Los esfuerzos de estandarización se han enfocado hacia los sistemas de *charging* basados en políticas [17][18][19]. Mediante el uso de sistemas basados en políticas, es posible intercambiar mensajes que contienen información de gestión agregada (incluida la información de tarificación), y, por lo tanto, reducir la señalización.

3.4. Descripción de la información usada en la tasación

Según el tipo de información a la que se refieren los datos de *charging*, la información se ha clasificado en los siguientes grupos: información del perfil de usuario,

información del modelo de tasación e información del *accounting record*.

La información de perfil de usuario o *user profile* se refiere a la información relacionada con el usuario final, gestionada por el PSP. Para efectos de la tasación, solamente se toma en cuenta la información usada para este proceso, por lo que se centra en una fracción de lo que se conoce como el perfil de usuario. Estos parámetros incluyen, por ejemplo, el monto del crédito que es disponible para gastar, la lista de los servicios a los cuales el usuario tiene derecho a acceder, potenciales límites de crédito para ciertos servicios, y referencias el modelo de tasación usado. El perfil de usuario es actualizado durante o después del proceso de tasación, por ejemplo, deduciendo el crédito gastado por el uso de un servicio, actualizando contadores.

Información del Modelo de Tasación (Charging Model) es el conjunto de datos contenidos en reglas que determinan, por un lado, cómo calcular el costo de un servicio y, por otro, las tarifas usadas.

Información de Accounting Record (generado por el proceso de *accounting*) es la colección de datos de los registros de uso agregados desde los diferentes elementos de red generalmente por el proceso de *metering*. Estos registros son generalmente usados en la tasación fuera de línea para propósitos de posprocesamiento, y en ese caso son llamados Charging Data Records. De manera general, son también usados en la tasación en línea como parámetros de entrada para el cálculo del costo de los servicios en tiempo real.

Una vez el servicio es iniciado, el proceso de *charging* solicita el perfil de usuario y el modelo de tasación convenido como datos de tasación iniciales e inicia el proceso de *charging*. En consecuencia, los procesos de *metering* y *accounting* son usados para generar los registros de *accounting* y los registros de los datos de uso, respectivamente. El registro de *accounting* es usado para calcular el costo del servicio en un cierto punto de la sesión de servicios, lo cual habilita al proceso de *charging* para permitir o denegar uso adicional del servicio. Durante o después del proceso de *charging*, el perfil de usuario se actualiza en consecuencia.

3.5. Charging en los estándares del 3GPP

Hay una gran cantidad de documentos del 3GPP que se ocupan de los procesos de la tasación en línea (*online charging*). La Especificación Técnica (TS) 32.240 [21] es el documento general que contiene una descripción de la estructura de las especificaciones de *charging* en el 3GPP (tanto en línea como fuera de línea). Las especificaciones son categorizadas en dos partes: la

primera cubre diferentes arquitecturas relacionadas con la tasación y la segunda se dedica a los parámetros de tasación, descripción de la sintaxis e interacciones dentro de la red.

El operador de red *home* del usuario (PSP) es definido por el 3GPP como una entidad central que es responsable por la tasación del usuario final. Si el PSP tiene acuerdos de *roaming* con otros operadores de red, el usuario estará habilitado para usar la infraestructura de red de esos operadores. En este caso, el sistema de tasación de PSP es también responsable de los procedimientos de *charging* interdominio. Además, el usuario final está en capacidad de consumir servicios proveídos por proveedores de aplicaciones y contenido de terceros.

Los principios y los requerimientos de tasación generales para la arquitectura del 3GPP son listados en el TS22.115 [21][22]. Los requerimientos más importantes son los siguientes:

- Los usuarios finales deben ser conscientes y conocer todos los cargos relacionados con ellos. Esto incluye el hecho de informar a los usuarios sobre las tasas que están a punto de ser aplicadas, para que ellos puedan aceptar o rechazar el servicio.
- Se debe incorporar la funcionalidad para tasar separadamente cada componente de medios dentro de una sola sesión para realizar la tasación de acuerdo a los recursos de red usados.
- Se debe considerar la función de tasar a los usuarios según su localización, su presencia, es decir, el contexto del servicio que está siendo consumido.
- El modelo de tasación acordado con el PSP debe respetarse, así este se encuentre en la red de otro operador usando las funciones del *roaming*.

De manera general la tasación *online* definida por el 3GPP se ve en la figura 4. La tasación puede ser usada a tres niveles dentro de la arquitectura: a nivel de transporte, a nivel de subsistema y a nivel de servicio. Las funcionalidades de tasación son ejecutadas en diferentes niveles. A nivel de transporte, *charging* puede controlar los recursos de red usados en el despliegue del servicio. A nivel del subsistema, donde se sitúa IMS, la

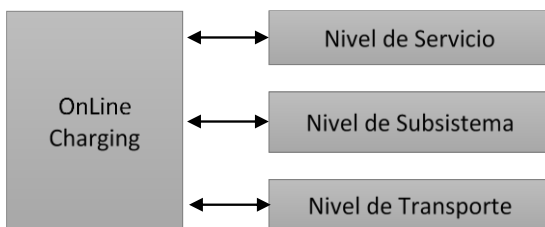


Figura 4. Niveles operativos de la tasación según el 3GPP. Fuente: elaboración propia.

tasación controla las sesiones de los servicios y permite o deniega el inicio de sesión. En el nivel de servicio, los servicios específicos pueden ser tasados (ej. IPTV), así como el contenido de un servicio en particular.

Un Sistema de Tasación en Línea (Online Charging System, OCS), definido en TS 32.296 por el 3GPP [23], es una arquitectura funcional que provee soporte a los tres niveles de la tasación en línea.

El OCS también apoya la arquitectura de la política y control de carga (Policy and Charging Control, PCC), la cual conecta los procesos a nivel de subsistema con los de nivel de transporte.

3.5.1. Antecedentes de los estándares IETF

La arquitectura AAA genérica descrita en el RFC 2903 especifica un *framework* que incorpora la autenticación de usuario, la autorización de servicio y los procedimientos de *accounting*, apuntados para ser usados en el entorno de internet. Aunque fue publicada en el año 2000, esta arquitectura aún se usa (parcial o completamente) como base en la construcción del estado del arte de las arquitecturas de *charging*, incluyendo al OCS del 3GPP. El 3GPP adoptó dos ideas claves de la arquitectura AAA para el entorno de las telecomunicaciones:

- Los procesos de *accounting* y, por lo tanto, de *charging* son llevados a cabo por un punto central en la red.
- Las funciones de *accounting* son incorporadas dentro de los procedimientos de autorización de servicio. La descripción detallada de los procedimientos de *accounting* definidos por la IETF se encuentran en el RFC 3334 [25].

El protocolo Diameter es el usado en el OCS del 3GPP para la transmisión de la información de AAA. Este se especifica en el RFC 2865 [26]. El protocolo Diameter básico contiene mensajes para las funcionalidades AAA en general, así como la especificación del contenido de esos mensajes, almacenados en estructuras de datos llamadas AVP (Attribute Value Pairs). Cualquier otro procedimiento que haya sido especificado para un servicio en particular o una función de red que usa Diameter, pueden definirse mediante aplicaciones de Diameter. Estas aplicaciones son la aplicación de control de crédito de Diameter, RFC 4006 [24][25], usado por el OCS de 3GPP, para controlar el costo durante la sesión del servicio, y las aplicaciones de interfaces específicas usadas para señalar diferentes funciones 3GPP.

El control de crédito es un mecanismo usado para interacciones en tiempo real entre un OCS y un proveedor de servicio para controlar o monitorear todas

las tasas relacionadas con su uso. El mecanismo puede ser usado para escenarios de tasación basados en eventos como en los basados en sesión.

4. Identificación y clasificación de la información de tasación usada en 3GPP OCS

La información de tasación usada en la tasación *online*, planteada por el 3GPP, se puede agrupar en dos grupos:

- 1) Información proveída por las funciones de red al OCS cuando se requiere tasación *online*, incluyendo el nivel de servicio, nivel de subsistema y las funciones de transporte.
- 2) Información almacenada en el OCS y usada en la tasación.

El primer grupo de información se muestra en la Tabla 1.

Tabla 1. Lista de información de tasación (3GPP TS 22.115)

Información proveída por el usuario final	Identidad Identidad de la red Home Identidad del terminal Recurso solicitado Parámetros de QoS (Quality of Service) Identificador de servicio solicitado
Información proveída por la red Home o una red en <i>roaming</i>	Identificación de la red Tiempo universal para eventos específicos durante la sesión Cantidad de datos transferidos desde y hacia el usuario Recursos asignados al usuario QoS proveído Localización del usuario Información de presencia
Información de tasación proveída por un proveedor de una tercera parte	Identidad del proveedor Tiempo universal para eventos específicos durante la sesión Tipo de servicio Tipo de contenido

En general, los actores que pueden verse incluidos en el despliegue de un servicio son los siguientes: el usuario final, la red Home u operador local, la red visitada (en caso de *roaming*) y proveedores de servicio de terceros (si el usuario está accediendo a ese servicio). Los datos, en su mayoría, contienen información de identidad, unidades de servicio solicitadas, QoS solicitado, localización e información de presencia.

El segundo grupo de información es almacenada en las funciones que pertenecen al OCS, y esta ha sido mencionada previamente. La información más importante incluye

- Crédito disponible del usuario final
- Contadores relacionados a la historia de uso del servicio del usuario final, número permitido de unidades de servicio, etc.
- La lista de los servicios activos por usuario final
- La lista de precios de los servicios

Los primeros ítems representan información relacionada con el usuario final: disponibilidad de crédito como una parte del perfil de usuario; ciertos contadores pueden tomar determinados valores dependiendo de la suscripción del usuario y una lista de los servicios activos que representan una parte del contexto en el cual el usuario final consume el servicio dado. Sin embargo, la información acerca de los servicios que son tasados es reducida a la información acerca del precio del servicio. Además, el OCS en sí mismo no provee ninguna especificación acerca de los procedimientos interdominio, esta surge de la información almacenada en los SLA establecidos entre los proveedores.

Finalmente, dadas las numerosas funciones de red que pueden conectarse con el OCS, haciendo uso de los puntos de referencia estandarizados, para proveer al OCS con la información de tasación necesaria y solicitudes de tasación, surge el problema de la gran cantidad de señalización generada por el *charging*.

5. Sincronización de bases de datos

Para reiterar lo expuesto, el estándar IMS es una solución efectiva y aceptada mundialmente para la consecución de la convergencia de redes y, por ende, de las NGN. Por este motivo, se enfatizó el análisis en la proposición de un mecanismo que se integre fácilmente en un entorno IMS.

Con el problema puntualizado a IMS y comprendidos los requerimientos de las NGN para la incorporación de un sistema de tasación adaptable, se procedió a analizar el estado del arte de las técnicas de replicación en las redes de datos.

En las redes de datos, la sincronización de información se puede ver como una replicación de las actualizaciones que se hayan presentado en un determinado momento en algunas bases de datos.

5.1. Replicación en bases de datos

La replicación en bases de datos puede clasificarse de acuerdo a dos aspectos fundamentales, el primero es en

qué momento se realiza la replicación y el segundo es quién es capaz de realizar las actualizaciones en las diversas réplicas [27].

Dependiendo del momento en el que se realiza la replicación esta puede clasificarse en *eager* (temprana) o *lazy* (perezosa) [28].

- Replicación *eager*: también conocida como replicación síncrona, consiste en que las actualizaciones son propagadas dentro del tiempo de ejecución de la transacción solicitada por el cliente, es decir, la transacción es recibida por un nodo en especial, y este la ejecuta localmente, luego envía las actualizaciones que hayan tenido lugar a las demás réplicas y queda en espera de una respuesta de estas para verificar que no existan problemas de inconsistencia, una vez garantizada la consistencia del sistema, el nodo que recibió la solicitud entrega una respuesta al cliente, en caso de encontrarse problemas de consistencia la transacción es abortada por completo y ningún cambio surte efecto en ninguna réplica [27][28][29][30].

- Replicación *lazy*: permite que las transacciones se realicen localmente en el servidor que recibió la petición e inmediatamente después de completadas una respuesta es enviada al cliente. La propagación de las actualizaciones hacia las demás réplicas únicamente es enviada después de completada la transacción. Este comportamiento ofrece un bajísimo tiempo de respuesta, algo que, sumado con la simpleza de la técnica, se toma como el mayor fuerte de esta replicación. Sin embargo, en contraste con las técnicas *eager*, al no existir una fase de acuerdo entre todas las réplicas, la consistencia de la información es el factor débil de este tipo de replicación, y, por ende, muchas veces es necesario hacer uso de técnicas de reconciliación, para corregir los errores presentados durante la replicación [27][29].

Ahora, de acuerdo con quien realiza las actualizaciones, la replicación puede clasificarse en *primary copy* (copia primaria) o *update-anywhere* (actualización en cualquier punto).

- Replicación *primary copy*: en este esquema todas las actualizaciones deben ser realizadas por una réplica primaria, y posteriormente son propagadas a las demás réplicas, esto implica que únicamente la réplica primaria está en capacidad de actualizar la información, por lo tanto, las demás son únicamente réplicas de lectura. Esto simplifica enormemente el control de la replicación, ya que únicamente un nodo es responsable de las actualizaciones. Sin embargo, al estar centralizado el control, se genera un cuello de botella para todo el sistema, además de una pobre tolerancia a fallos [29].

- Replicación *update-anywhere*: al contrario del caso anterior, en estas técnicas las actualizaciones de la información pueden ser llevadas a cabo por cualquier replica, sin necesidad de pasar por ningún intermediario, esto hace que el sistema tenga un acceso mucho más rápido a la información, pero de igual forma implica que los procesos de coordinación de la réplicas sea mucho más complejo, y la posibilidad de que se generen conflictos entre distintas transacciones es mucho mayor [27].

A partir de la anterior clasificación, se puede obtener una combinación de técnicas según el momento en que se aquella se realiza y de quién realiza la replicación. La combinación resultante puede apreciarse en la figura 5.

Propagación de las actualizaciones

Localización de las actualizaciones	Eager Primary Copy	Lazy Primary Copy
	Eager Update Anywhere	Lazy Update Anywhere

Figura 5. Replicación en bases de datos.

Fuente: elaboración propia.

5.2. Técnicas de replicación

5.2.1. Middleware

Middleware es un *software* diseñado para ayudar a manejar la complejidad y la heterogeneidad inherente de los sistemas distribuidos. Este se define como una capa *software* que se encuentra ubicada sobre el recurso que se desea manejar y por debajo de la aplicación que lo manejará. Para lograr lo anterior, un *middleware* provee herramientas para los programadores, tales como librerías y API, lo que reduce significativamente las labores de desarrollo y evita de alguna forma que se cometan errores [31].

Típicamente existen dos alternativas para ofrecer replicación en bases de datos, la primera es realizar extensiones al código fuente del *core* del DBMS (Database Management System, Sistema de Gestión de Bases de datos), y la otra es hacer uso de una capa *middleware*. La primera ofrece un alto desempeño debido al poco *overhead* generado, pero es una solución completamente dependiente del proveedor del motor de base de datos, lo cual es una limitante en términos de interoperabilidad, portabilidad, migración y escalabilidad.

La segunda alternativa ofrece independencia del proveedor del DBMS, lo cual facilita su interoperabilidad con otros entornos y facilita la portabilidad de aplicaciones, además de su migración a versiones más recientes de los motores de bases de datos usados. Así pues, se podría pensar que estas ventajas compensan el elevado *overhead* producido por este tipo de soluciones.

Además de las mencionadas anteriormente, los sistemas *middleware* presentan una serie de ventajas que los convierte en una muy buena opción como sistema de replicación de bases de datos, entre las cuales se destacan las siguientes [31]:

- Compatibilidad para interactuar con sistemas heredados (*legacy*), esto dada la capacidad inherente de este tipo de sistemas para trabajar en entornos heterogéneos, lo que permite realizar una fácil migración y adaptación de aplicaciones y servicios que no fueron concebidos para interoperar con otros, por este motivo estos sistemas algunas veces son conocidos tecnología “pegante”.
- Facilidad de programación, esto significa que los desarrolladores de aplicaciones no necesitan realizar cambios ni aprender nuevos lenguajes de programación para utilizar estos sistemas dado que la mayoría proporcionan librerías con las funcionalidades que proveen, las cuales están escritas en lenguajes conocidos como son Java o C++.
- Distribución en capas, esto significa que pueden existir múltiples capas *middleware* en un sistema dado, esto permite maximizar los beneficios que aportan estos sistemas, así por ejemplo puede existir una capa que proporcione sincronía a través de un servicio de entrega atómica y a su vez esta sea utilizada por otra capa encargada de proveer tolerancia a fallos o balance de carga.
- Gestión de la calidad de servicio (QoS, Quality of Service), al ser el dinamismo una característica propia de los sistemas de bases de datos distribuidas, la programación alrededor de estos se torna compleja, es por esta razón que se busca hacer una abstracción de los requerimientos de QoS de las aplicaciones y llevarlos a un bajo nivel, donde sistemas gestores de recursos garantizan el correcto funcionamiento del sistema completo. Las capas *middleware* se adaptan perfectamente a este diseño, permitiendo a los desarrolladores enfocarse en la lógica de las aplicaciones o servicios que están implementando y delegar la complejidad de la gestión de la QoS a las capas inferiores.

5.2.2. Group communication

Una de las técnicas más utilizadas en los sistemas distribuidos es la Llamada a Procedimientos Remotos (RPC, Remote Procedures Call), que oculta el envío de los mensajes a otro nodo de la red y hace parecer que el procedimiento se está ejecutando localmente. Esta solución es eficiente, pero está diseñada para trabajar en entornos punto-punto, sin embargo, existen muchas aplicaciones y servicios que necesitan establecer comunicaciones punto-multipunto, lo que implicaría un elevado número de mensajes si se quisiera utilizar RPC en estos casos. Por ejemplo, si necesita comunicarse con n nodos en la red, entonces serían necesarios $n-1$ mensajes con sus respectivos ACK (*acknowledgement*), por lo que en total se estarían enviando $2(n-1)$ mensajes.

Además del *overhead* producido al utilizar únicamente comunicaciones punto-punto, el garantizar la consistencia en sistemas de bases de datos distribuidas es una tarea compleja, ya que para este fin es necesario que los mensajes tengan un orden y lo conserven al llegar a cada uno de los nodos [32].

A partir de lo anterior, surgen nuevas posibilidades de comunicación mediante el uso de protocolos que implementen primitivas de *group communication*, que puede ser visto como una forma de proveer comunicación multipunto-multipunto, organizando los procesos que se desean comunicar en grupos, por ejemplo, un grupo puede consistir en un conjunto de usuarios que desean establecer una partida de un juego en línea. Cada grupo tiene asociado un nombre lógico, así cuando un proceso desea comunicarse con el grupo lo hace enviando un mensaje al nombre que tenga asignado el grupo.

Además de reducir el *overhead*, este tipo de protocolos garantiza que los mensajes estén totalmente ordenados, lo cual es uno de los requisitos básicos que se tienen al momento de realizar replicación en bases de datos [31][32][33].

Entre las principales primitivas que hacen que los protocolos que implementan *group communication* sean una alternativa eficiente para la replicación se encuentran las siguientes [33]:

- Confiabilidad: permite a esta clase de protocolos brindar una comunicación fiable. Es en este punto donde recae el peso de la recuperación de fallos en la comunicación, tales como *overflow* en los *buffers* y el manejo de los paquetes erróneos o corruptos.
- Ordenamiento: esto hace referencia en la forma en la que los mensajes son enviados al grupo. Comúnmente existen cuatro formas de realizar este proceso: sin orden,

ordenamiento FIFO (*first in, first out*, primero en entrar, primero en salir) ordenamiento causal y ordenamiento total. El ordenamiento FIFO garantiza que todos los mensajes enviados por un miembro del grupo son recibidos en el mismo orden en que este los envió. El ordenamiento causal garantiza que todos los mensajes que un miembro envíe y los que estén relacionados con éste se entregarán en el mismo orden. Mientras que en el ordenamiento total se garantiza que todos los miembros del grupo reciben todos los mensajes enviados por cualquier otro en el mismo orden para todos los casos. Esta última alternativa es la más conveniente al momento de pensar en replicación, sin embargo, y como es de esperarse, es la más compleja de implementar.

- **Semántica de entrega:** hace referencia al momento en que un mensaje se considera que fue entregado satisfactoriamente al grupo. Para considerar esto, existen tres opciones: k-entregas, quórum y entrega atómica, en el primer caso se considera que el mensaje fue entregado cuando ha sido recibido por un número k de miembros, en el segundo como su nombre lo indica se considera entregado cuando la mayoría de los miembros del grupo han recibido el mensaje y finalmente la entrega atómica se refiere a que únicamente se considera que el mensaje fue entregado cuando todos los miembros del grupo reciban el mensaje.

Esta última opción es la ideal para ser usada en el caso de la replicación en bases de datos, ya que esto permite cumplir con las propiedades ACID (*Atomicity, Consistency, Isolation, Durability* - durabilidad, consistencia, aislamiento, durabilidad) de las transacciones que se verán en la sección.

Tradicionalmente, la replicación de bases de datos ha sido manejada de dos formas, ya sea mediante técnicas de replicación *eager* o *lazy*, sin embargo cada una de estas presenta sus propios problemas [28][29].

Si bien la replicación *lazy* es rápida y eficiente, debido a que las actualizaciones se realizan localmente y sólo se envían a las réplicas cuando la transacción ha sido completada; no garantiza una buena consistencia entre las réplicas lo cual se convierte en su principal problema. Por otro parte, las técnicas *eager* ofrecen una buena consistencia entre las réplicas, pero esto a costa de un alto tiempo de respuesta debido al uso de protocolos de acuerdo “atómico”, lo cual puede resultar en la generación de cuellos de botella [30][31].

Actualmente el incremento en el uso de sistemas de bases de datos distribuidos ha promovido el surgimiento de nuevos protocolos y técnicas de replicación para bases de datos basadas en primitivas de comunicación de grupo (*group communication*), las cuales a través de

propiedades tales como el ordenamiento y la atomicidad de los mensajes ofrecen enormes ventajas a este nuevo tipo de protocolos eliminando de esta manera la posibilidad de formar cuellos de botella, además de reducir la carga de mensajes en el sistema, lo cual genera un incremento en el desempeño sin llegar a comprometer la consistencia de la información [31].

Otra de las opciones que se presenta al momento de realizar la replicación de bases de datos es el uso de arquitecturas *middleware*, que pueden ser configuradas dependiendo de las características del entorno donde van a ser aplicadas, para lograr de esta forma hacer más énfasis en ciertos aspectos como lo son la escalabilidad y la consistencia de la información. El uso de estas arquitecturas trae consigo la gran ventaja de lograr una independencia del fabricante de las bases de datos que se desean replicar, lo cual inherentemente garantiza la interoperabilidad entre los diferentes Sistemas de Gestión de Bases de Datos (*DataBase Management System, DBMS*) y la portabilidad de aplicaciones [30]

6. Propuesta del mecanismo de sincronización de información de tasación en IMS

Para seleccionar una solución al problema planteado tanto a nivel general de las NGN, como a nivel particular en IMS, se deben tener en cuenta algunas condiciones como la facilidad de implementación, los costos que representa y que la solución sea lo menos intrusiva posible con los estándares ya establecidos.

Además, las NGN presentan requerimientos que son determinantes al momento de realizar una investigación en estas redes entre los que se encuentran el uso eficiente de los recursos (principalmente el ancho de banda), la constitución de una red funcionalmente distribuida y basada en IP y un sistema abierto de protocolos internacionalmente aceptado.

También se añade el hecho de que se maneja un número muy elevado de usuarios, por lo que se debe garantizar una alta escalabilidad y un desempeño aceptable del sistema sin importar la carga de este.

Además, al analizar los requerimientos se encontró que las primitivas de *group communication*, usadas en las técnicas de replicación de los DDDBS (Distributed Database Management System, Sistema de Gestión de Bases de Datos Distribuidas), y las arquitecturas *middleware* ayudan a cumplir esos de la siguiente forma:

- Las técnicas de replicación que usan *group communication* ayudan a lograr un transporte oportuno de la información, ya que reducen el *overhead* en los procesos de replicación [11].

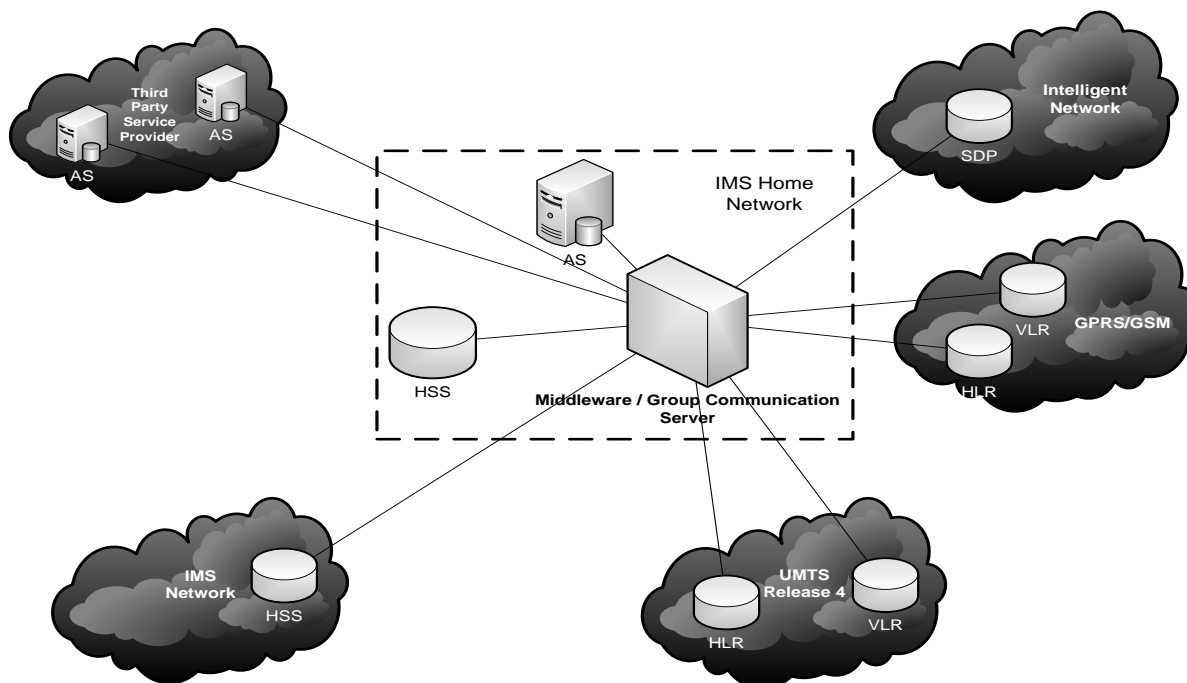


Figura 6. Incorporación de la plataforma *middleware* con *group communication* en IMS.

Fuente: elaboración propia.

- *Middleware* ofrece independencia del proveedor del DBMS, lo que facilita su interoperabilidad en entornos de bases de datos heterogéneos y heredados. Además, facilita la portabilidad de aplicaciones, y ayuda a que la solución sea poco intrusiva y fácil de implementar [33].
- El uso de *group communication* en un mecanismo de replicación garantiza la consistencia de la información gracias a las primitivas que usa [32].
- Una arquitectura *middleware* con *group communication* permite un uso eficiente de los recursos de red, ya que reduce considerablemente el *overhead* [33].
- El manejo de un número elevado de usuarios no es un problema con las arquitecturas *middleware*, ya que la escalabilidad es una propiedad inherente de ellas [34].
- Una arquitectura *middleware* está pensada para trabajar en entornos geográficamente distribuidos [33].

En la figura 6 se puede apreciar el diseño inicial para la incorporación de la plataforma *middleware* con *group communication* en IMS. Una plataforma especializada se encarga del proceso de replicación mediante las técnicas de los DDBS y las bondades ofrecidas por las arquitecturas *middleware*, interconectando las bases de datos que sean necesarias para compartir en tiempo real la información que necesitan.

Entonces, dependiendo de las exigencias, puede replicarse información entre bases de datos de la misma red IMS, con otras redes IMS, con redes de proveedores de servicios o con redes heredadas.

Del análisis de la información anterior se lograron las siguientes deducciones:

- Por las características de su arquitectura, se puede determinar que IMS presenta una similitud con los DDBS, ya que las bases de datos de cada red y dominio existentes comparte información entre ellas. Por lo tanto, se considera viable la aplicación de técnicas de replicación de los DDBS a IMS.
- Estas bases de datos, por pertenecer físicamente a diferentes operadores de red y proveedores de servicio, son heterogéneas en ciertas características, como el DDBMS que usan (Oracle, PostgreSQL, MySQL), la funcionalidad de sus servidores (HLR, Home Location Register; HSS, VLR, Visitor Location Register, etc.) y los protocolos e interfaces implementadas en estos.
- En IMS existen varias bases de datos que se pueden considerar heredadas, esto debido a la interoperabilidad con otros tipos de redes (GPRS/GSM, General Packet Radio Service; UMTS, Universal Mobile Telecommunications System; Red Inteligente; RDSI,

Red Digital de Servicio Integrados; PSTN, Public Switched Telephone Network) [35].

- El core IP en IMS es de gran importancia para una eventual interconexión de estas bases de datos.
- El HSS, como repositorio central de la información en IMS, tiene varias tareas asignadas, por lo que es bastante útil delegar la funcionalidad de replicación de información a otro módulo cuando esta labor requiera un alto consumo de recursos.

Así, se determinó que la forma de mejorar el desempeño en los procesos de replicación de bases de datos en IMS, de modo que se garantice la consistencia en su información, era introducir las técnicas de replicación usadas en el contexto de las redes de datos, en especial en los sistemas de bases de datos distribuidos (DDBS). Esto se debe principalmente a la posibilidad de ver las bases de datos de IMS como un DDBS y a la opción de usar IP como protocolo base para las conexiones necesarias.

Posteriormente, se examinaron los requerimientos y se encontró que las primitivas de *group communication* usadas en las técnicas de replicación de los DDBS, junto con las arquitecturas *middleware*, ayudan a cumplirlos de la siguiente forma:

- Las técnicas de replicación que usan *group communication* ayudan a lograr un transporte oportuno de la información, ya que reducen el *overhead* en los procesos de replicación [30][36].
- *Middleware* ofrece independencia del proveedor del DBMS, lo que facilita su interoperabilidad en entornos de bases de datos heterogéneos y heredados. Además, facilita la portabilidad de aplicaciones, y ayuda a que la solución sea poco intrusiva y fácil de implementar [13].
- El uso de *group communication* en un mecanismo de replicación garantiza la consistencia de la información, gracias a las primitivas que usa [36][38].
- Una arquitectura *middleware* con *group communication* permite un uso eficiente de los recursos de red, ya que reduce considerablemente el *overhead* [36][37].
- El manejo de un número elevado de usuarios no es un problema con las arquitecturas *middleware*, ya que la escalabilidad es una propiedad inherente de ellas.
- Una arquitectura *middleware* está pensada para trabajar en entornos geográficamente distribuidos.

En la figura 6 se puede apreciar el diseño inicial para la incorporación de la plataforma *middleware* con *group communication* en IMS. Una plataforma especializada se encarga del proceso de replicación usando las técnicas de los DDBS y las bondades ofrecidas por las arquitecturas *middleware*, interconectando las bases de datos que sean necesarias para compartir en tiempo real la información que necesiten.

Por ejemplo, en el caso de la tasación del servicio IPTV, el *core* de la red IMS contará con la información registrada por el HSS

7. Prototipo para la validación del mecanismo propuesto

7.1. Diseño

De acuerdo con lo analizado entre los módulos de la Arquitectura de Servicios Nativos de IMS, se comparte cierta información que puede ser estática o puede variar durante una sesión establecida. Por lo tanto, para realizar la implementación del prototipo dentro de esta arquitectura se debe seleccionar un tipo de información dinámica y que pueda ser modificada fácilmente.

Además, dado que el mecanismo se ha planteado para la sincronización de información entre bases de datos, debe seleccionarse dos módulos que tengan bases de datos en su estructura.

Entonces, se determinó probar el mecanismo propuesto, sincronizando los nodos que comparten la información del *Repository Data* para un servicio dado dentro del IMS. Estos nodos son el HSS y los AS que prestan el servicio tal como se especifica en [34].

El HSS y los AS tienen bases de datos que pueden ser conectadas a la plataforma *middleware*, y, por otra parte, la información del *Repository Data* es de gran importancia para la prestación de servicios personalizados, ya que los proveedores de servicio pueden ingresar en estos campos cualquier tipo de dato relacionado con los servicios que un usuario en particular consume [6].

De esta forma, se plantea la inclusión de una plataforma *middleware* entre los nodos mencionados. La arquitectura del prototipo para la validación se muestra en la figura 7.

Sin el ánimo de plantear una modificación específica al estándar IMS, sino con la finalidad de realizar las pruebas del mecanismo propuesto, se inhabilita la interfaz Sh y, por ende, su propio mecanismo de sincronización.

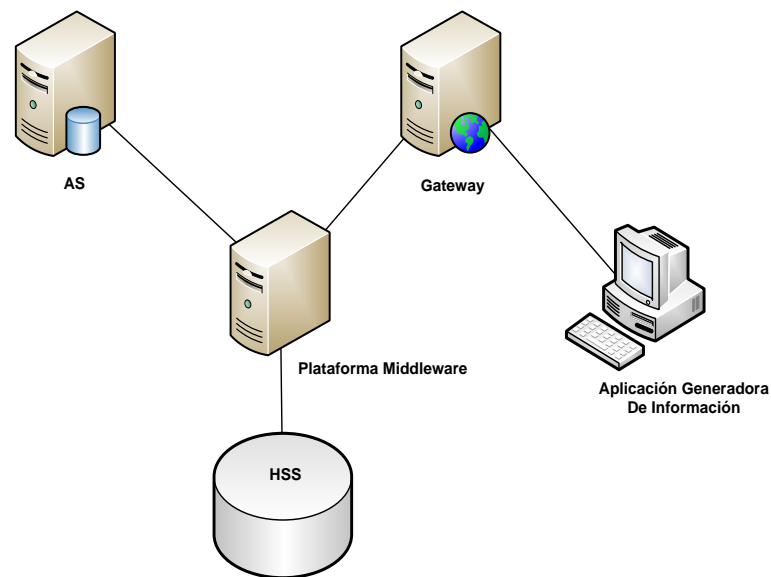


Figura 7. Arquitectura general del prototipo. Fuente: elaboración propia.

Además de los elementos mencionados anteriormente, se decide realizar la inclusión de un AS que actúa como *gateway* entre la aplicación que genera la información a ser sincronizada y la plataforma de replicación.

7.1.1. Implementación y pruebas

Para la validación de la propuesta, se plantea la simulación de un servicio en la Arquitectura de Servicios Nativos IMS, el cual genera y requiere información del Repository Data.

El servicio simulado es el de localización de un usuario móvil y de lugares de interés en mapas que se descargan desde el AS al cliente IPTV.

El AS se encarga de actualizar la ubicación del usuario de acuerdo a la información suministrada por una Aplicación Generadora de Coordenadas (AGC), que emula los datos generados por el cambio de posición del equipo terminal.

La AGC genera las coordenadas para cada usuario, las cuales se guardan en la base de datos del HSS (específicamente en el Repository Data) y en la base de datos del AS y son actualizadas cada vez que se genere otra actualización usando un mecanismo de sincronización de información.

Se plantearon dos escenarios, en el primero se usó el mecanismo de sincronización a través de la interfaz Sh

de IMS, y en el segundo se usó el mecanismo propuesto empleando una plataforma *middleware*. Como ya se había dicho anteriormente, se realizaron pruebas con estos dos mecanismos con el objetivo de tener un punto de referencia que permitiera evaluar los resultados obtenidos.

Con cada mecanismo se realizaron tres capturas de los mensajes que se transfirieron entre los nodos que intervienen en el proceso. En cada captura se mantuvo fijo el número de actualizaciones en 50, pero se varió el número de usuarios para el cual la AGC está actualizando la información de localización y el retardo entre cada actualización, después de recibir un acuse de recibo en la AGC. Esto se hizo con el ánimo de variar la carga de la red en cuanto a procesamiento y número de *bytes* por segundo que se transporta por la misma.

El número de usuarios se varió entre 10, 50 y 100, mientras que los retardos para cada captura fueron 500 ms, 250 ms y 125 ms. Así, la captura con menos carga para la red tuvo 10 usuarios y un retardo de 500 ms entre cada actualización, y la captura con más carga para la red tuvo 100 usuarios y un retardo de 125 ms.

Para cada uno de los casos que se presentan, se tomaron estadísticas del tiempo que se tardó cada mecanismo en completar todas las actualizaciones y el número de *bytes* que se generaron y se transportaron por en la red.

Además, el analizador de protocolos Wireshark suministró estadísticas de otros parámetros, que se describen a continuación.

-Tiempo entre el primer y el último paquete (secs): es el tiempo total que se tarda el mecanismo en procesar todas las peticiones y en enviar los mensajes a los nodos que correspondan.

-Paquetes: es el número total de paquetes generados por todos los nodos que intervienen en el mecanismo.

-Promedio paquetes/seg: es el promedio de paquetes generados que se transportan por la red en un segundo.

-Promedio tamaño de paquetes (bytes): es el promedio del tamaño de todos los paquetes generados.

-Bytes: es el número de bytes total generados por todos los nodos que intervienen en el mecanismo.

-Promedio bytes/seg: es el promedio de bytes generados que se transportan por la red en un segundo.

-Promedio Mbit/seg: es el promedio de megabits de datos que se transportan por la red en un segundo, lo cual indica la carga promedio de la red en un segundo.

El análisis de los resultados obtenidos se enfocó en la observación del comportamiento de los dos mecanismos en cuanto al tiempo de respuesta y a la carga que se genera en la red. Lo anterior se debe a que, como se expuso con la propuesta, se pretende mejorar el desempeño de la red en la sincronización de información y garantizar al mismo tiempo consistencia en ella.

El tiempo promedio de procesamiento para todas las capturas en el escenario A se condensan en la tabla 2.

Tabla 2. Promedio del tiempo de procesamiento de las peticiones de actualización en el escenario A (resultados en segundos)

		Retardo de actualización		
		500 ms	250 ms	125 ms
n.º de usuarios	10	0,56846	0,6386	0,73676
	50	2,40148	2,51248	2,9131
	100	5,14264	5,68542	6,06312

En la tabla 3 se compilan los valores del tiempo promedio de procesamiento de un grupo de peticiones de actualización en el escenario B. Se observa que los valores se encuentran en casi todos los casos por debajo del valor de retardo de actualización, lo que quiere decir que los componentes de mecanismo alcanzan a procesar el grupo de peticiones de actualización antes de que se genere otro, con lo que se garantiza la sincronización oportuna y consistente de la información.

Para complementar la validación del prototipo del mecanismo de sincronización propuesto, se consideró pertinente realizar una comparación con el mecanismo de sincronización de IMS usando la interfaz Sh, con el propósito de verificar el comportamiento de la propuesta frente a una implementación que ya se encuentra definida para una arquitectura de las NGN.

Para cada uno de los retardos de actualización se muestran los resultados obtenidos del número de bytes total y tiempo de respuesta con su respectiva comparación.

El tiempo de respuesta se refiere al tiempo que le toma al sistema realizar la replicación de los datos. Con los valores presentados en las tablas de la 4 a la 9, se evidencia un mejor comportamiento del mecanismo propuesto respecto al mecanismo propio de IMS, y una mejoría en su respuesta en el tiempo en hasta 16 veces para el caso de mayor exigencia (100 usuarios con actualizaciones cada 125 ms). Además, es importante notar que la carga adicional a la red (información de cabeceras y su procesamiento) se ubica por debajo en número de bytes generados, cuando se usa el sistema de replicación de información.

Tabla 3. Promedio del tiempo de procesamiento de las peticiones de actualización en el escenario B (resultados en segundos)

		Retardo de actualización		
		500 ms	250 ms	125 ms
n.º de usuarios	10	0,06402	0,06304	0,02944
	50	0,182	0,1632	0,12234
	100	0,40864	0,30252	0,24502

Tabla 4. Comparación en *kbytes* entre los mecanismos con un periodo de actualización de 500 ms

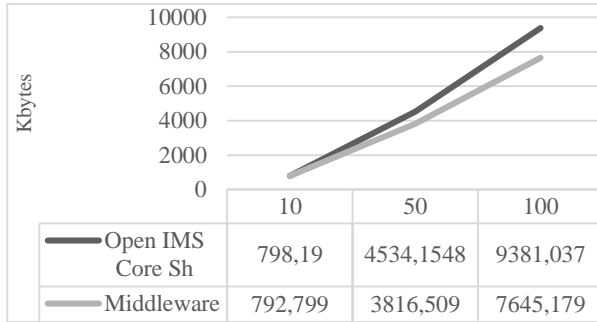


Tabla 5. Comparación en tiempo entre los mecanismos con un periodo de actualización de 500 ms

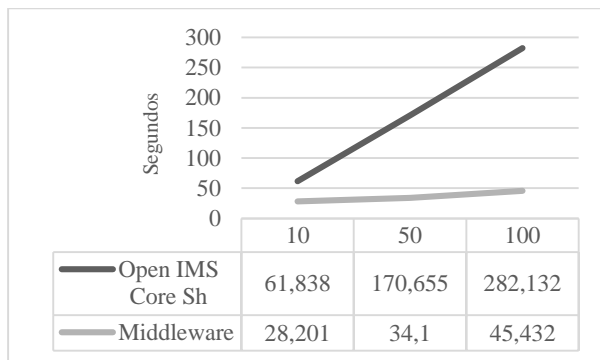


Tabla 6. Comparación en *kbytes* entre los mecanismos con un periodo de actualización de 250 ms

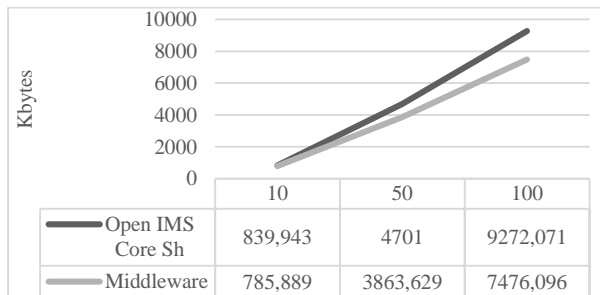
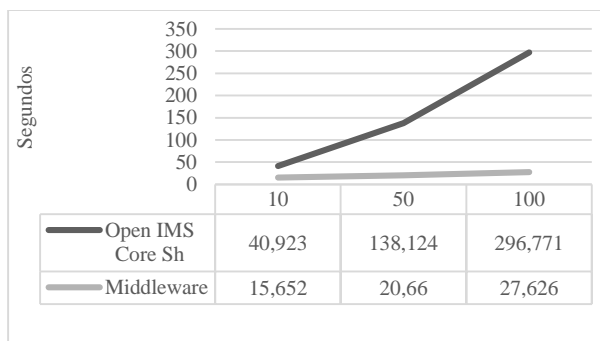


Tabla 7. Comparación en tiempo entre los mecanismos con un periodo de actualización de 250 ms



A pesar de que estas arquitecturas *middleware* introducen *overhead*, se observa que la propuesta presentada genera aproximadamente un 20 % menos de *bytes* que el mecanismo de IMS, mientras se ofrece una considerable mejora en la respuesta en el tiempo y se garantiza la consistencia de la información.

Los datos muestran también que el desempeño del mecanismo propuesto es superior. Esto se evidencia especialmente en la medida en que la exigencia para el sistema sea mayor; en el caso de que aumente el número de peticiones en el tiempo, la carga para los componentes aumenta, pero el desempeño del mecanismo propuesto es mucho mejor frente al mecanismo de IMS, lo que corrobora las ventajas de las arquitecturas *middleware* en entornos donde se requiere de una alta escalabilidad.

Tabla 8. Comparación en *kbytes* entre los mecanismos con un periodo de actualización de 125 ms

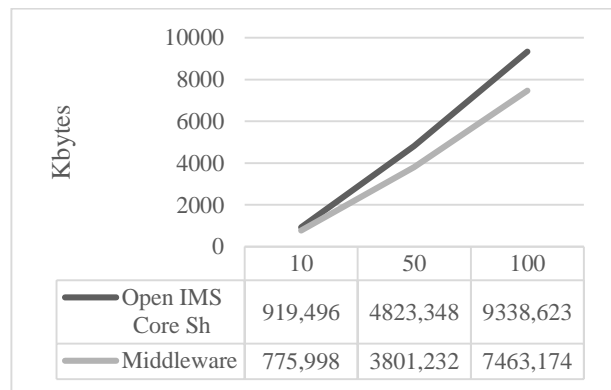
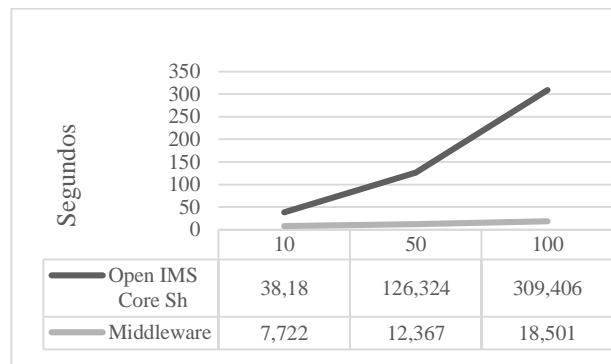


Tabla 9. Comparación en tiempo entre los mecanismos con un periodo de actualización de 125 ms



8. Conclusiones

La sincronización de la información de usuario en el entorno IMS es susceptible de mejoras en cuanto al desempeño de la red y la consistencia de los datos, si se hace uso de las propuestas provenientes del campo de los

Sistemas de Bases de Datos Distribuidos (DDBS, Distributed DataBase Systems).

La incorporación a IMS de una plataforma *middleware*, que implemente *group communication* y que esté especializada en la sincronización de datos, permite un uso más eficiente de los recursos de red, y al mismo tiempo provee la facilidad de integración con los repositorios de datos de redes heredadas.

Para el mecanismo de sincronización propuesto, el *overhead* introducido de forma inherente por hacer uso de una plataforma *middleware* es aceptable si se realiza una comparación del tráfico que se produce al hacer uso del mecanismo propuesto de IMS.

La introducción de una plataforma *middleware* como mecanismo de sincronización en el entorno de IMS no implica la realización de cambios drásticos en la configuración de los componentes de red, por lo que es viable realizar su implementación en un entorno real.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo de la Universidad del Cauca, Colombia, específicamente del Grupo de Ingeniería Telemática, y al programa de Maestría en Ingeniería Telemática de la Universidad del Cauca. Además, ha sido parcialmente financiado por Redauti (Red Temática en Aplicaciones y Usabilidad de la Televisión Digital Interactiva), financiado por CYTED.

Referencias

[1] Q. Quboa and N. Mehandjiev, "Creating Intelligent Business Systems by Utilising Big Data and Semantics," *IEEE 19th Conference on Business Informatics (CBI)*, Thessaloniki, 2017, pp. 39-46. doi: 10.1109/CBI.2017.71

[2] B. O. Obele, S. H. Han, J. K. Choi and M. Kang, "On building a successful IPTV business model based on personalized IPTV content & services," *9th International Symposium on Communications and Information Technology*, Icheon, 2009, pp. 809-813. doi: 10.1109/ISCIT.2009.5341129

[3] T. Itälä and H. Töhönen, "Difficult Business Models of Digital Business Platforms for Health Data: A Framework for Evaluation of the Ecosystem Viability," *IEEE 19th Conference on Business Informatics (CBI)*, Thessaloniki, 2017, pp. 63-69. doi: 10.1109/CBI.2017.6

[4] C. Bormann, S. Flake, y J. Tacke, "Business Models for Local Mobile Services Enabled by Convergent Online Charging", en *Advances in Mobile*

and Wireless Communications, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 281-296. doi: 10.1007/978-3-540-79041-9_15

[5] S. Forge, "The future of global telecommunications in view of the growth of OTT services: expected impacts on usage and prices," *ITU/BDT Regional Economic and Financial Forum of Telecommunications/ICTs for Latin America and the Caribbean*, Nassau, Bahamas, April, 2015.

[6] G. Camarillo and M. A. Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*. Wiley, 2012.

[7] *Service requirements for the Internet Protocol (IP) multimedia core network subsystem, Stage 1, TS 22.228*, 3rd Generation Partnership Project (3GPP).

[8] T. Russell, *The IP Multimedia Subsystem (IMS): Session Control and Other Network Operations*. USA: McGraw-Hill, 2008.

[9] *IP Multimedia Subsystem (IMS); Stage 2 (Release 8)*, 3GPP Technical Specification 23.228 Versión 13.5.0, 2015.

[10] S. A. Ahson and M. Ilyas, *IP Multimedia Subsystem (IMS) Handbook*. New York, USA: Taylor & Francis Group, 2009.

[11] *Network architecture, Release 12, TS 23.002*, 3rd Generation Partnership Project (3GPP).

[12] M. Wiesmann, A. Schiper, "Comparison of Database Replication Techniques Based on Total Order Broadcast," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 551-566, 2005. doi: 10.1109/TKDE.2005.54

[13] S. Karnouskos, "Mobile payment: A journey through existing procedures and standardization initiatives," in *IEEE Communications Surveys & Tutorials*, vol. 6, no. 4, pp. 44-66, Fourth Quarter 2004. doi: 10.1109/COMST.2004.5342298

[14] C. A. Gizelis and D. D. Vergados, "A Survey of Pricing Schemes in Wireless Networks," in *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 126-145, First Quarter 2011. doi: 10.1109/SURV.2011.060710.00028

[15] H. Hakala, L. Mattila, J. P. Koskinen, M. Stura, J. Loughney, *Diameter credit-control application*, RFC 4006, 2005.

- [16] *Vocabulary for 3GPP specifications*, 3rd Generation Partnership Project (3GPP).
- [17] P. Kurtansky, B. Stiller, "State of the art prepaid charging for IP services," in *4th Wired/Wireless Internet Communications*, 2006, vol. 3970, pp. 143-154. doi: 10.1007/11750390_13
- [18] Z. Ezziane, "Charging and pricing challenges for 3G systems," in *IEEE Communications Surveys & Tutorials*, vol. 7, no. 4, pp. 58-68, Fourth Quarter 2005. doi: 10.1109/COMST.2005.1593280
- [19] M. Koutsopoulou, A. Kaloxylou, A. Alonistioti, L. Merakos and K. Kawamura, "Charging, accounting and billing management schemes in mobile telecommunication networks and the internet," in *IEEE Communications Surveys & Tutorials*, vol. 6, no. 1, pp. 50-58, First Quarter 2004. doi: 10.1109/COMST.2004.5342234
- [20] T. Trygar and G. Bain, "A framework for service level agreement management," *MILCOM 2005 - 2005 IEEE Military Communications Conference*, Atlantic City, NJ, 2005, pp. 331-337 Vol. 1. doi: 10.1109/MILCOM.2005.1605706
- [21] *Charging and accounting principles for NGN*, ITU-T Recommendation D.271, 04/2008
- [22] *Service aspects; charging and billing*, 3GPP TS 22.115, Rev. 11.5.0, 2011
- [23] *Telecommunication management; charging management; online charging system (OCS): Applications and interfaces*, 3GPP TS 32.296, Rev. 11.2.0, 2011.
- [24] *Service Aspects; Service Principles*, 3GPP Technical Specification 22.101 Versión 7.9.0, 2007.
- [25] T. Zseby, S. Zander, G. Carle, *Policy-based accounting*, RFC 3334, 2002.
- [26] C. Rigney, S. Williams, A. Rubens, W. Simpson, *Remote authentication dial in user service (RADIUS)*, RFC 2865, 2000.
- [27] B. Kemme, R. Jimenez, M. Patiño-Martínez, *Database Replication*. Morgan&Claypool Publishers, 2010. doi: 10.2200/S00296ED1V01Y201008DTM007
- [28] B. Kemme, R. Jiménez-Peris, M. Patiño-Martínez, G. Alonso, "Database Replication: A Tutorial", en *Lecture Notes in Computer Science*, Springer., 2010, pp. 219–252.
- [29] S. Kumar, G. Urmit, D. Singh, "Analysis of a database replication algorithm under load sharing in networks," *Journal of Engineering Science and Technology*, vol. 11, núm. 2, pp. 193–211, 2016.
- [30] S. Dayyani, M. Reza-Khayyambashi, "A Comparative Study of Replication Techniques in Grid Computing Systems," *International Journal of Computer Science and Information Security*, vol. 11, núm. 9, pp. 64-73, 2013
- [31] D. Bakken, "Middleware," *Encyclopedia of Distributed Computing*, Kluwer Academic Press, 2003 [En línea]. Disponible en: <http://www.eecs.wsu.edu/~bakken/middleware-article-bakken.pdf>.
- [32] M. T. Özsu y P. Valduriez, *Principles of Distributed Database Systems, Third Edition*. New York, NY: Springer New York, 2011. doi: 10.1007/978-1-4419-8834-8
- [33] B. Kemme, "Database Replication Based on Group Communication: Implementation Issues", en *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2003, pp. 132–136. doi: 10.1007/3-540-37795-6_24
- [34] *Technical Specification 23.008 Organization of Subscriber Data*, 3GPP [En línea]. Disponible en: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=731>
- [35] Fraunhofer Institute FOKUS, "The Open IMS Core Project," 2008. [En línea]. Disponible en: <http://www.openimscore.org>
- [36] GORDA Project, "State of the Art in Database Replication," 2006. [En línea] Disponible en: <http://gorda.di.uminho.pt/library/wp1/GORDA-D1.1-V1.2-p.pdf>
- [37] J. E. Armendáriz-Iñigo, H. Decker, F. D. Muñoz-Escóí, y J. R. G. de Mendivil, "A Closer Look at Database Replication Middleware Architectures for Enterprise Applications", en *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2007, pp. 69–83. doi: 10.1007/978-3-540-75912-6_6
- [38] "Continuent.org Sequoia 3.0 Basic Concepts". [En línea] Disponible en: <http://sequoia.continuent.org/doc/latest/sequoia-basic-concepts.pdf>.