Aplicación de Multicast IPv6 Seguro a Servicios de Información en Entornos Grid

Departamento de Arquitectura de Computadores y Automática

Universidad Complutense de Madrid

Natalia Bibiana Trejo

Director: Juan Carlos Fabero Jiménez

Proyecto de Fin de Máster en Investigación en Informática

Madrid, Julio de 2008

Resumen

Los Servicios de Información constituyen piezas fundamentales de la infraestructura de los sistemas Grid. Estos servicios tienen por objetivo realizar el descubrimiento inicial y posterior monitorización de la disponibilidad y estado de los recursos y servicios puestos a disposición por los participantes de las llamadas Organizaciones Virtuales. El servicio de información utilizado por sistemas Grid basados en el middleware Globus Toolkit 4, se conoce como Sistema de Monitorización y Descubrimiento.

Por otra parte, GridWay es un *metaplanificador*, una herramienta de planificación y gestión de ejecución de trabajos integrada a Globus Toolkit 4, que permite compartir a gran escala y de manera fiable y eficiente recursos de cómputo gestionados por diferentes sistemas Gestores de Recursos Locales. GridWay consulta, entre otros, al Sistema de Monitorización y Descubrimiento de Globus Toolkit 4 para obtener información de los recursos Grid y realizar la planificación de ejecución de trabajos.

Por último, las propuestas relacionadas con la incorporación de multicast a Grid lo implementaron en el nivel de capa de aplicación o utilizando TCP y principalmente a los servicios de transferencia de ficheros en grid computacionales y de datos o en la compartición de aplicaciones multimediales. Estas propuestas se basaron en el protocolo IPv4, a pesar del soporte para la transmisión multicast IPv6 que comunica los nodos de la red multicast troncal de Internet, M6BONE, y de la compatibilidad de la mayoría de los middleware Grid con el protocolo IPv6.

El objetivo del presente trabajo es presentar los resultados de la aplicación de multicast IPv6 al descubrimiento y monitorización de recursos y servicios en sistemas Grid basados en GT4. Se diseñó un modelo de organización de servicios de información que transmiten sus datos sobre multicast IPv6 de manera segura utilizando certificados digitales. Este nuevo esquema de organización de servicios de información permite que los servicios Índices de MDS4 se estructuren de manera plana, descentralizada, redundante y tolerante a fallas. De esta forma GridWay pue-

de planificar la ejecución de los trabajos accediendo a información más actualizada y que se encuentra disponible de forma redundante en servicios de información pertenecientes a un grupo multicast.

La integración de multicast IPv6 que utiliza certificados digitales con los Servicios de Información, permite que la información relativa a la disponibilidad y estado de los recursos de un sistema Grid, se distribuya de manera redundante y segura y que los tiempos de retardo sean menores que si se implementa el mismo modelo mediante transmisión unicast.

Agradecimientos

En primer a mi esposo y a toda mi familia, por su incondicional cariño y apoyo en esta empresa que he iniciado hace dos años.

En segundo lugar, a Fundación Carolina por haber confiado en mis capacidades académicas para continuar estudiando en España.

Quisiera agradecer a mi tutor, por su paciencia e ideas a la hora de realizar este trabajo y a todas las personas del Dpto. que me han hecho sentir muy cómoda en el día a día.

En cuarto lugar a toda la gente amiga, que me ha aconsejado técnica y personalmente en todo momento, desde aquellas que se encuentra muy lejos hasta las que están cerca, por ejemplo en el laboratorio. A todos y todas, *gracias totales*.

Finalmente, agradecer al proyecto CyTED 506PI0293 para el que he colaborado con este trabajo.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Introducción Conceptual y Motivación

La tecnología Grid surge como un nuevo paradigma de computación distribuida. Propuesta por Ian Foster y Carl Kesselman a mediados de los 90 [18, 19], se basa fundamentalmente en el acceso remoto a recursos distribuidos, y su principal objetivo es permitir gestionar recursos de diversos tipos, tales como datos, almacenamiento, servicios, redes, sensores, clusters, etc., de tal forma que los usuarios se beneficien de ellos a pesar de que se encuentren dispersos geográficamente y pertenezcan a diferentes organizaciones, logrando alcanzar de forma segura y económica capacidades computacionales que permitan la ejecución eficiente de aplicaciones intensivas en datos o computación.

Existen numerosos proyectos científicos y empresariales que hacen uso de la tecnología Grid con excelentes resultados, por ejemplo Compute Against Cancer, GriPhyN, EGEE-II, SETI@home, proyectos de Novartis o BBC, entre otros. La relevancia de esta nueva tecnología también se pone de manifiesto a través del desarrollo de proyectos a nivel iberoamericano tales como el proyecto *Tecnología Grid como motor del desarrollo regional* [9], perteneciente al Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo, cuyos objetivos son variados desde el tratamiento de imágenes médicas, aplicaciones en el campo de la biología molecular hasta sistemas de prevención y gestión de riesgos naturales, desarrollo de sistema de soporte a la decisión clínica, entre otros.

Se ha definido un estándar o modelo de referencia para la tecnología Grid conocido como *Open Grid Services Architecture (OGSA)* [35]. En esta arquitectura (Fig. 1.1), el cuello de botella reside en las capas de Recursos y Conectividad. En particular la capa de Conectividad utiliza la pila de protocolos TCP/IP para

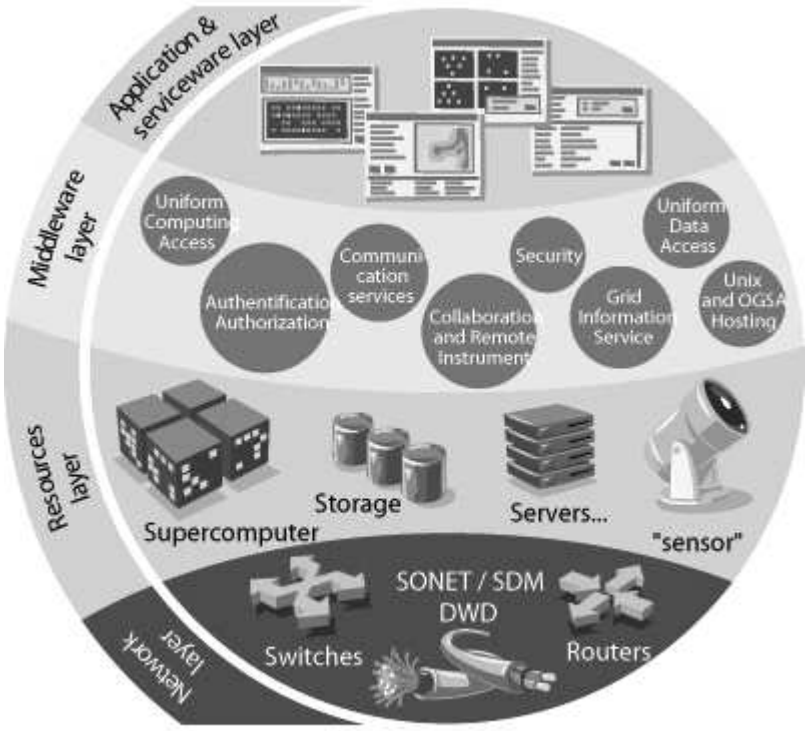

Figura 1.1: Arquitectura Grid [23].

establecer la comunicación en una red Grid. De esta manera, el desarrollo de la tecnología Grid se ve afectada por la propia evolución de Internet y el que, por ahora, es uno de los cambios más importantes que experimenta la Red: la sustitución del protocolo IPv4 por el IPv6.

El protocolo IPv6 [11] nace con el propósito de mejorar las prestaciones de IPv4 en cuanto a seguridad y rendimiento, dar soporte a los nuevos tipos de servicios requeridos y superar el problema del agotamiento de direcciones disponibles. Este cambio se encuentra en fase de implantación y es un proceso prolongado durante el cual ambas versiones del protocolo IP deben coexistir. Actualmente IPv6 ya está funcionando en muchos centros de investigación, no sólo en el plano experimental. La red de investigación de la Comunidad de Madrid, RediMadrid, ofrece soporte nativo para el protocolo IPv6 y las siguientes instituciones ya se han decidido a emplearlo: UAM, UPM, UC3M y CSIC.

El objetivo de este trabajo es desarrollar nuevas aplicaciones para la tecnología Grid que permitan aprovechar las características en cuanto a comunicación multicast

que ofrece IPv6. Principalmente se propone estudiar la mejora en el descubrimiento de los recursos y servicios disponibles en un sistema Grid al emplear las posibilidades de multicasting de IPv6.

1.2. Estado del Arte

Hasta ahora todos los productos de software para la construcción de sistemas Grid han sido desarrollados sobre IPv4 y, aunque su funcionamiento sobre la nueva versión del protocolo está garantizado, son muchos los estudios que enfatizan sobre la conveniencia del desarrollo de Grids sobre IPv6 o que por lo menos sean independientes de la versión IP que utilizan [34].

La filosofía que subyace en el concepto de Grid es la posibilidad de compartir todos los recursos disponibles a nivel mundial siguiendo estructuras cliente-servidor pero también la comunicación directa peer-to-peer entre usuarios. Esta filosofía se enfrenta directamente con el problema del escaso número de direcciones potenciales que ofrece IPv4 frente a IPv6. Otro de los obstáculos que encuentra la tecnología Grid es la implantación de NAT (Network Adress Translation), nacida con la finalidad de ahorrar direcciones públicas en IPv4 ya que dificulta la conectividad y acceso a los recursos.

La implementación de IPv6 aporta soluciones para superar estos obstáculos descritos a la vez que ofrece nuevos beneficios:

- Gran cantidad de direcciones, que hará virtualmente imposible que queden agotadas.
- Direcciones unicast, multicast y anycast.
- Formato de cabecera más flexible que en IPv4 para agilizar el encaminamiento.
- Nueva etiqueta de flujo para identificar paquetes de un mismo flujo.
- La fragmentación se realiza en el nodo origen y el reensamblado en los nodos finales, y no en los encaminadores como en IPv4.
- Nuevas características de seguridad, IPsec es nativo para IPv6, y por tanto, la encriptación y autenticación se implementa a nivel de paquete.
- Autoconfiguración de los nodos finales, que permite a un equipo aprender automáticamente una dirección IPv6 al conectarse a la red.

- Movilidad incluida en el estándar, que permitirá cambiar de red sin perder la conectividad.

Globus Toolkit (GT4) [16] ha emergido como el estándar de facto para la capa intermedia (middleware) del Grid (Fig. 1.2), permite compartir recursos localizados en diferentes dominios de administración, con diferentes políticas de seguridad y gestión de recursos. GT4 incluye, entre otros, servicios que permiten:

1. La gestión de recursos a través del servicio de Gestión y Asignación de Recursos en Grid (*Grid Resource Allocation and Management, GRAM*)
2. La monitorización y descubrimiento de información mediante el Sistema de Descubrimiento y Monitorización (*Monitoring and Discovery System, MDS4*)
3. La gestión y movimiento de datos a través del servicio de FTP en Grid (*GridFTP*)
4. La implementación de comunicaciones seguras mediante la Infraestructura de Seguridad Grid (*Grid Security Infrastructure, GIS*)

La mayoría de los sistemas Grid que se expanden sobre las comunidades académicas, de investigación y empresariales están basados en la herramienta Globus Toolkit como núcleo de la capa intermedia, además la última versión de Globus Toolkit ya cuenta con soporte para IPv6.

Uno de los principales servicios Grid con posibilidades de mejora al utilizar IPv6, es el Sistema de Monitorización o Descubrimiento.

Para comprender las mejoras que puede aportar IPv6 a este fin necesitamos comprender algunos conceptos básicos relacionados con la propagación de paquetes mediante el protocolo TCP/IP: unicast, anycast y multicast.

- Unicast: es un esquema de direccionamiento donde existe una asociación unívoca entre el emisor y el receptor por lo que el emisor debe enviar una copia de los datos a cada uno de los receptores.
- Anycast: es un esquema de direccionamiento por el cual los datos son dirigidos al destino más próximo o *mejor de los destinos*. La dirección destino identifica a un grupo de posibles receptores pero solo uno de ellos es finalmente elegido como destinatario de la información basándose en diferentes criterios de encaminamiento.

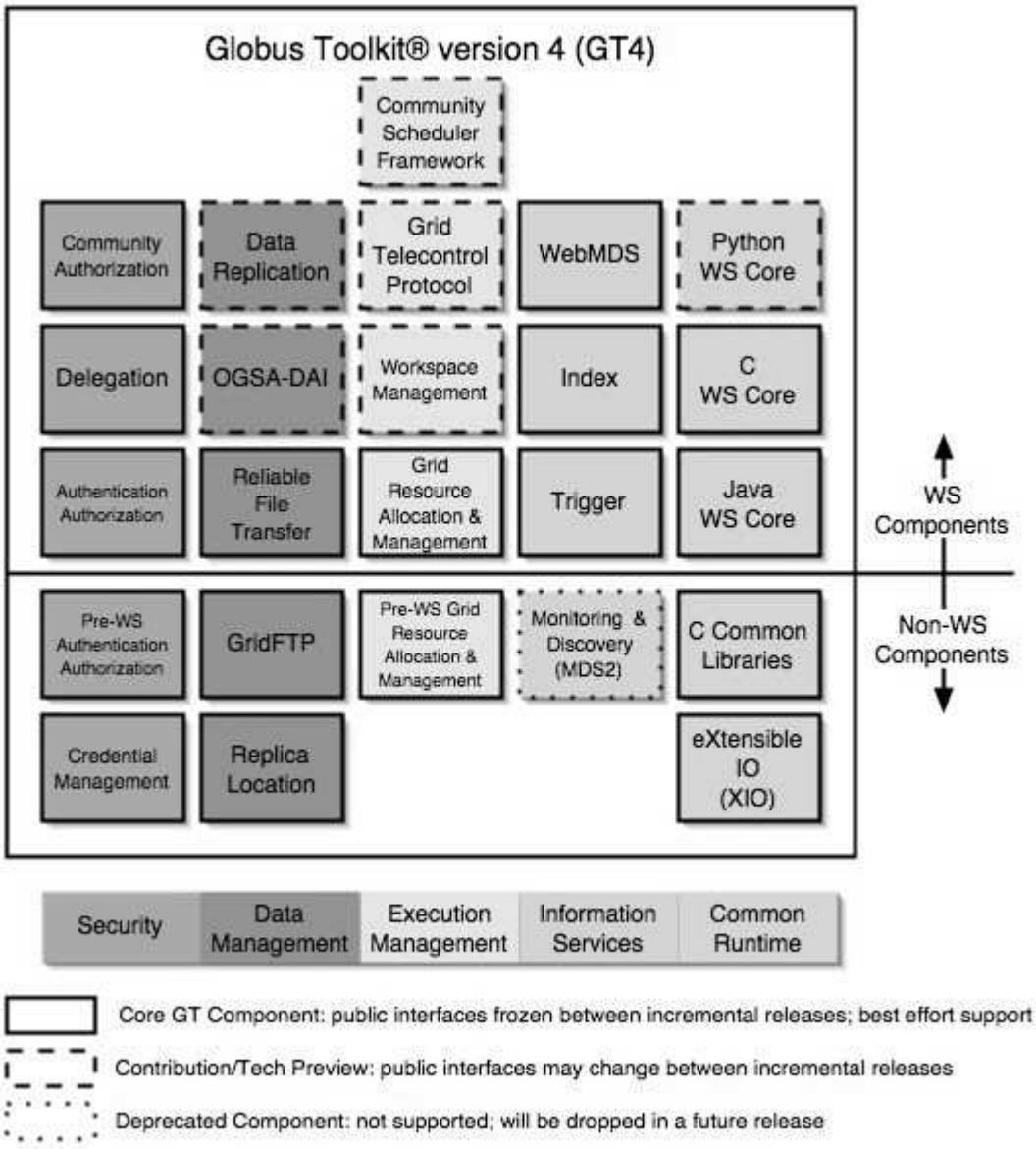

Figura 1.2: Componentes de Globus Toolkit 4 [43].

- Multicast: es un esquema de direccionamiento donde la asociación entre emisores y receptores es uno a muchos. El emisor envía una sola copia y es la red quien dirige los datos a cada uno de los receptores.

El metaplanificador GridWay [25] como herramienta incluida en GT4, debe obtener la información acerca de la infraestructura Grid subyacente para llevar a cabo la planificación de recursos. GridWay selecciona aquellos recursos más adecuados para la ejecución de los trabajos luego de consultar diferentes Sistemas de Información, entre ellos los servicios de MDS4, concretamente, los servicios Índices de MDS4.

El presente proyecto tiene como finalidad profundizar en el estudio de la combinación de la tecnología Grid con multicast IPv6, comprobando que el descubrimiento y monitorización de recursos y servicios disponibles en los sistemas Grid y a los que accede el metaplanificador GridWay, mejora al utilizar las posibilidades de multicasting de IPv6.

1.3. Objetivos Cumplidos y Metodología Empleada

1.3.1. Objetivo General

El objetivo de este trabajo consiste en el diseño, implementación y evaluación experimental de un nuevo esquema de organización de servicios Índices de MDS4, que aproveche las ventajas del multicast IPv6 transmitiendo información de manera segura a cada uno de los servicios MDS4 pertenecientes a un grupo multicast dentro una Organización Virtual (VO).

1.3.2. Objetivos Específicos

- Identificar de qué manera los Servicios de Información Grid son susceptibles de ser mejorados mediante el uso de multicast IPv6
- Diseñar y desarrollar la solución de software que implemente dichas mejoras
- Desplegar la solución en los nodos de prueba
- Obtener valores preliminares de referencia que permitan realizar optmizaciones y ajustes para luego desplegar la solución entre *sites* geográficamente distantes

1.3.3. Metodología

En general, se pretende diseñar y evaluar soluciones de software que implementen multicast IPv6, midiendo tiempos de ejecución sobre un escenario simulado lo más cercano a la situación real.

- Estudio del funcionamiento del Sistema de Monitorización y Descubrimiento de GT4, del metaplanificador GridWay y de multicast IPv6.
- Estudio de la implementación de aplicaciones multicast IPv6 que soporten firmas digitales.
- Implementación de un esquema de organización de servicios Índices para MDS4 que distribuya la información necesaria para GridWay sobre multicast IPv6. La implementación, a la vez, permite utilizar las credenciales X.509 proporcionadas por la Infraestructura de Seguridad Grid de GT4 para validar integridad y autenticidad de la información transmitida sobre multicast IPv6.
- Comparación cuantitativa mediante tiempos de ejecución sobre los nodos de prueba, tanto de la solución que incluye seguridad como de la solución que incorpora seguridad a través del soporte para certificados digitales.

1.4. Organización del Proyecto

El trabajo está organizado de la siguiente forma:

Capítulo 2 Se describe el funcionamiento del Sistema de Monitorización y Descubrimiento de Gt4; se describen los componentes de MDS4 involucrados en el desarrollo de este proyecto.

Capítulo 3 Se describen el funcionamiento del Metaplanificador GridWay y su interacción con los Servicios de Información.

Capítulo 4 Se describen el estado del arte de multicast IPv6, el modelo de comunicación, aplicaciones, encaminamiento multicast, protocolos de encaminamiento e implementación de multicast IPv6.

Capítulo 5 Se describen los trabajos relacionados con la integración de multicast a sistemas Grid.

Capítulo 6 Se describe el diseño general del modelo de organización de servicios Índices MDS4 basado en multicast IPv6. Luego se describen los modelos seguros y no seguros y cómo se transmite información a un conjunto de MDS4 pertenecientes a un grupo multicast a los que puede consultar GridWay.

Capítulo 7 Se describe de la implementación de las soluciones diseñadas propuestas en el Capíutlo 6.

Capítulo 8 Se describen los experimentos y resultados obtenidos de las soluciones diseñadas propuestas en el Capíutlo 6 .

Capítulo 9 Se presentan las conclusiones de este trabajo teniendo presente los resultados obtenidos.

Capítulo 2

Servicios de Información en Globus Toolkit 4

2.1. Sistema de Monitorización y Descubrimiento

Los recursos que un sistema Grid pone a disposición de la VO son naturalmente cambiantes. Por una parte, nuevos recursos y servicios se incorporan al sistema Grid (nuevos servidores de ficheros, nuevos gestores de réplicas, nuevos gestores de recursos locales, etc.), y por otra parte recursos y servicios existentes son quitados o se convierten en inaccesibles. Asimismo el estado de cada recurso, representado por los valores de sus propiedades, puede ser tanto estático como dinámico; ejemplo de propiedades estáticas incluyen el tipo de sistema operativo, ancho de banda, velocidad de CPU, etc., y ejemplos de propiedades cambiantes comprenden el consumo de memoria, tiempo de CPU, uso de ancho de banda, nodos disponibles para recibir trabajos en un clúster, etc. [39].

- El *Descubrimiento* es el proceso de encontrar los recursos adecuados para realizar una tarea, por ejemplo, encontrar un host en el que ejecutar un *job*. Este proceso puede involucrar encontrar qué recursos son los más adecuados (por ejemplo, tener la arquitectura de CPU correcta) y elegir el miembro adecuado en ese conjunto (por ejemplo, el nodo de un clúster que tenga la cola de envío más corta).
- La *Monitorización* es el proceso de observación de los recursos o servicios (por ejemplo computadores y planificadores) para realizar un seguimiento de su estado o para resolver problemas. Por ejemplo, un usuario puede usar un sistema

de monitorización para identificar recursos que están agotando el espacio en disco y en consecuencia tomar medidas correctivas.

El Sistema de Monitorización y Descubrimiento de GT4 (*Monitoring and Discovery System, MDS4*) [42], basado en las especificaciones definidas en Web Service Resource Framework (WSRF) y WS-Notification (WS-N) [17], es un conjunto de servicios web que permite que todos los recursos y servicios de un sistema Grid puedan ser descubiertos y monitorizados de una manera uniforme.

Por lo tanto, los servicios y recursos computacionales Grid pueden anunciar una gran cantidad de datos para diferentes propósitos. MDS4 fue diseñado especialmente para permitir que estos datos se encuentren disponibles para múltiples personas en múltiples dominios administrativos o *sites*. No obstante, no debe confundirse con un sistema de monitorización de sistemas distribuidos, como NetLogger, o con un monitor de clúster, pues MDS4 puede comunicarse con estos sistemas y con archivos de monitorización de maneras más detalladas, para luego publicar un resumen de esos datos utilizando interfaces estándares.

MDS4, también conocido como *WS MDS (Web Service MDS)*, se encuentra integrado por los siguientes servicios (ver Fig. 2.1):

- *Servicio Índice* (*Index Service*), servicio que recopila y publica información de los recursos y servicios del sistema Grid.
- *Servicio Trigger* (*Trigger Service*), servicio que recopila información de los recursos y ejecuta acciones especificadas por el usuario (por ejemplo, enviar un e-mail o generar una entrada en un fichero de logs) cuando se cumplen ciertos criterios (por ejemplo, cuando se agota el espacio en disco, o el servidor alcanza cierto límite).
- *Aggregator Framework*, framework de software sobre el que se construyen los servicios anteriores.
- *Proveedores de Información*, servicios que recolectan información mediante las llamadas *fuentes agregadoras* (en el Aggregator Framework).
- *WebMDS*, *front-end* que permite a los usuarios consultar los datos del servicio Índice a través de una interfaz web.
- *UsefulRP*, componente de software extensible usado para generar dinámicamente valores XML para uno o más WSRF Resource Properties en cualquier servicio compatible con GT4 Java WSRF-Core, disponible a partir de GT4.0.5+.

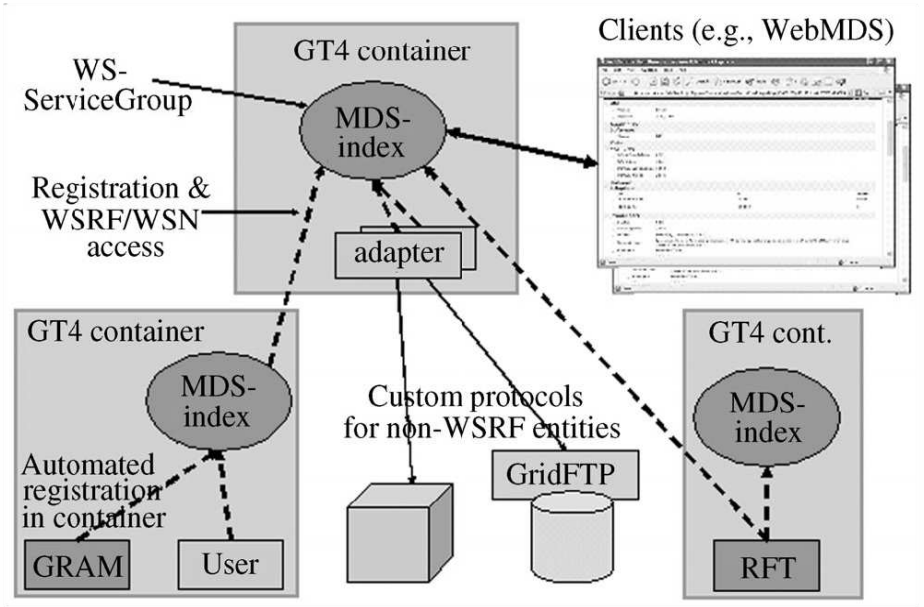

Figura 2.1: Monitorización y Descubrimiento en GT4.

- *Archive Service*, servicio, aún en desarrollo, que mantiene datos sobre los recursos y servicios de una forma persistente para que el usuario pueda consultar información histórica.

En las siguientes secciones describiremos aquellos componentes de MDS4 que puede ser beneficiados con el empleo de la tecnología multicast IPv6.

2.2. Servicios Índices de MDS4

El *servicio Índice* de MDS4 recolecta información referida a los recursos Grid y luego publica esos datos a través de *resource properties* (un conjunto de información ese recurso). Asimismo, almacena tanto la ubicación de recurso como una versión de copia de los datos del recurso que se mantiene por un período de tiempo.

Un sistema Grid maneja múltiples servicios Índices: cada contenedor GT4 tiene un servicio Índice por defecto que registra los recursos creados dentro de él. Cualquier servicio GRAM, RFT o CAS que se encuentra ejecutándose en ese contenedor se registra por a sí mismo al servicio Índice por defecto del contenedor. Además los *sites* y VOs mantienen uno o más servicios Índices para registrar los contenedores, recursos y servicios disponibles.

En general, se pueden configurar los servicios Índices de una manera libre, que puede ser jerárquica o no, dependiendo de las decisiones del administrador, y, aunque es muy frecuente hallar estructuras jerárquicas de servicios Índices en los sistemas Grid, no existe un único Índice global que provea información acerca de cada recurso Grid.

2.3. Aggregator Framework

Aggregator Framework (Fig. 2.2) es un framework de software usado para construir servicios que recopilan y agregan datos. Tanto el *servicio Índice* como el *servicio Trigger* se construyen sobre este framework, se conocen como servicios agregadores (*aggregator services*) y tienen las siguientes características en común:

- Recopilan información a través de *Fuentes Agregadoras (Aggregator Sources).* Un *Aggretator Source* es una clase de Java que implementa una interfaz para recopilar datos en formato XML.
- Usan un mecanismo de configuración común para mantener la información acerca de qué Aggregator Source usar y sus parámetros asociados (que especifican qué datos obtener y desde dónde hacerlo).
- Implementan un modelo de consistencia de datos donde el administrador es el que determina la frecuencia con que los datos publicados se renuevan, de esta manera la sobrecarga a raíz de las actualizaciones se reduce a expensas de tener datos ligeramente más antiguos.
- Asocian a cada registro un tiempo de vida, es decir que si un registro expira sin haber sido actualizado, esa entrada y sus datos asociados son eliminados del servidor. De esta forma, las entradas obsoletas son eliminadas automáticamente cuando sus registros dejan de ser renovados. Esta característica se conoce como *self-cleaning.*

MDS4 incluye los siguientes tipos de *Aggregator Sources*:

- *Query Aggregator Source*, que recopila periódicamente información de un recurso registrado utilizando mecanismos de consulta de tipo *WS-Resource Properties.*
- *Subscription Aggregator Source*, que recopila información de un recurso registrado a través de mecanismos WS-*Notification*, los datos son enviados cuando los valores de la propiedad del recurso se modifican.

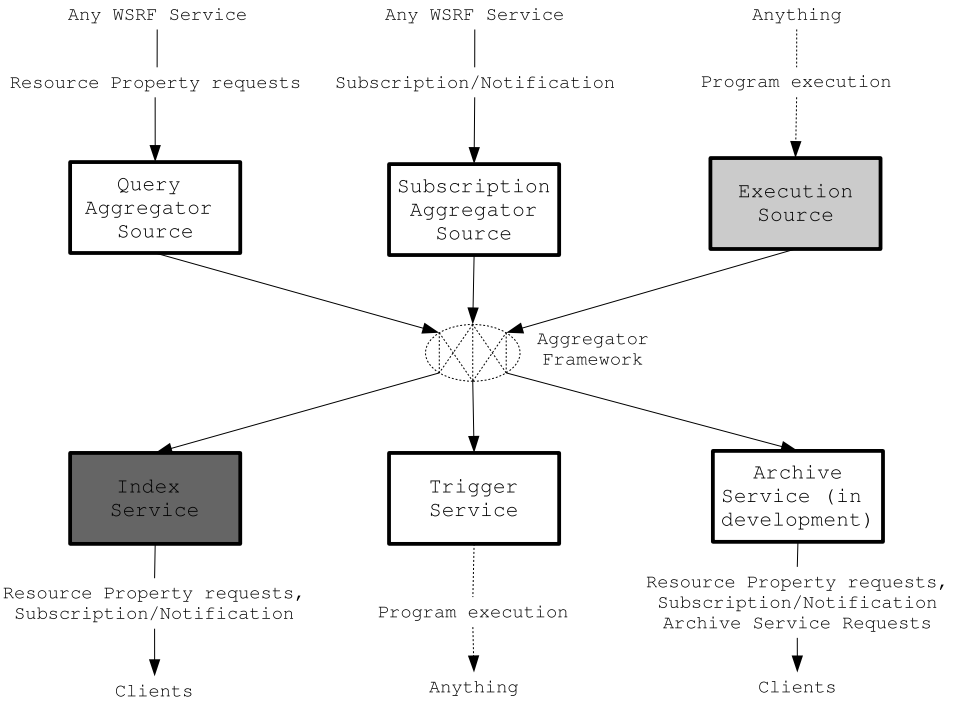

Figura 2.2: Flujo de Información en MDS4.

- *Execution Aggregator Source*, recopila información a partir de la ejecución de un programa suministrado por el administrador y devuelve un documento en formato XML.

Como puede observarse el servicio Índice recopila información en formato XML a través de estas *fuentes agregadoras*, sin embargo dichas fuentes, por defecto, no utilizan credenciales de autenticación sino que recuperan la información usando autenticación SSL anónima o ninguna autenticación.

2.4. Proveedores de Información

Los datos que las fuentes agregadoras de MDS4 publican en un servicio agregador se obtienen de un componente externo, llamado *Proveedor de Información*. En el caso de fuentes agregadoras de Consulta o Suscripción, el proveedor de información es un servicio compatible con WSRF del que se obtienen los datos mediante mecanismos WS-ResourceProperty o WS-Notification. En el caso de una fuente agregadora de Ejecución, el proveedor de información es un programa ejecutable que obtiene datos usando mecanismos específicos de ese programa.

Algunos proveedores de información que incluye MDS4 son los siguientes:

1. *Hawkeye Information Provider* y *Ganglia Information Provider*: proveedores de información que recogen datos de los recursos de un *pool* de Condor y de un clúster mediante Ganglia, respectivamente, generando información XML mediante un mapeo basado en el esquema GLUE [22] y reportándolo al servicio WS GRAM, el que los publica como *resource properties*. Esta información incluye:
   - nombre e id del host
   - información del procesador
   - tamaño de memoria
   - nombre y versión del sistema operativo
   - datos del sistema de ficheros
   - carga del procesador
2. *WS GRAM*: el servicio, componente de GT4, para el envío de trabajos. Este servicio WSRF publica información relativa al planificador local, incluyendo:
   - información de cola
   - número de CPUs disponibles y libres
   - cantidad de trabajos
   - estadísticas sobre el consumo de memoria
3. *RFT (Reliable File Transfer Service)*: el servicio, componente de GT4, para la transferencia de archivos. Este servicio WSRF publica:
   - estado de los datos del servidor
   - estado de la transferencia de uno o varios ficheros
   - número de trasferencias activas
   - información acerca de los recursos donde se ejecuta el servicio
4. *CAS (Community Authorization Service)*: estos servicios WSRF publican información que identifica la VO a la que sirven.
5. Cualquier otro servicio WSRF que publica *resource properties*.

Capítulo 3

Servicios de Información y Metaplanificador GridWay

Los *jobs* son trabajos computacionales que pueden ejecutar operaciones de entrada/salida y mientras se ejecutan pueden afectar el estado del recurso computacional y sus sistemas de ficheros asociados. Tales *jobs* pueden requerir el traslado coordinado de datos hacia el recurso antes de su ejecución y el traslado de los resultados desde el recurso al finalizar su ejecución. Algunos usuarios, particularmente en jobs interactivos, utilizan los datos de salida mientras el job se ejecuta. En este contexto, la monitorización consiste en consultar tanto la información de estado del recurso así como la información relacionada con los cambios en el estado del job. Los recursos Grid normalmente son operados bajo el control de un planificador que implementa políticas de asignación y priorización mientras se optimiza la ejecución de todos los jobs para lograr eficiencia y rendimiento.

Concretamente, *WS GRAM* (*Web Services Grid Resource Allocation and Management*) [10] es el componente de GT4 que comprende un conjunto de servicios web compatibles con WSRF, cuyo objetivo es localizar, enviar, monitorizar y cancelar jobs sobre los recursos de un sistema Grid. WS GRAM es un conjunto de servicios y clientes que, mediante un protocolo común, comunica un amplio rango de diferentes planificadores o gestores de trabajos para clústeres y en lotes. La implementación de WS GRAM incluye interfaces para planificadores o sistemas Gestores de Recursos Locales (*LRM*), como *Sun Grid Engine (SGE)*, *Condor*, *Portable Batch System (PBS)*, *Load Sharing Facility(LSF)*, etc. Además los servicios WS GRAM están pensados para gestionar aquellos jobs con determinados requisitos, por ejemplos aquellos donde es importante que las operaciones sean confiables, donde interviene la gestión de credenciales o donde deba existir coordinación de ficheros, etc.

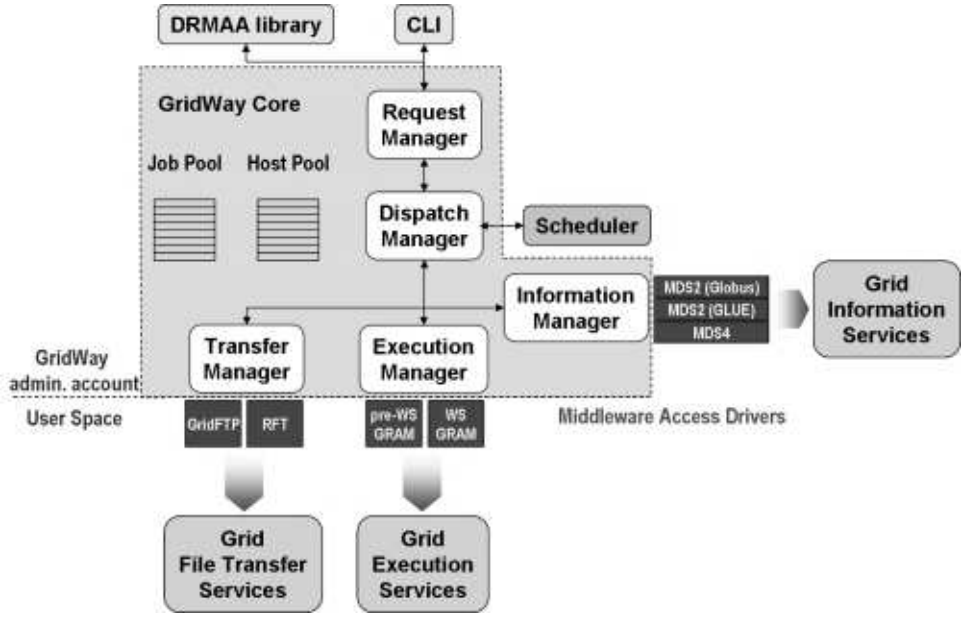

Figura 3.1: Componentes del Metaplanificador GridWay [12].

De esta forma surge el concepto de planificación Grid que consiste en encontrar una adecuada asignación entre jobs y recursos computacionales, considerando diferentes dominios administrativos y la heterogeneidad, dinamismo y control limitado que existe sobre los recursos.

GridWay [25] es un metaplanificador que permite la ejecución de trabajos, *array de trabajos* o trabajos complejos en un Grid dinámico y heterogéneo de forma eficiente y fiable. Es un metaplanificador que se utiliza con GT4 para la gestión de ejecución de trabajos que permite compartir a gran escala recursos de cómputo (clusters, servidores, supercomputadores) gestionados por diferentes *LRMs*, ubicados dentro de una misma organización o dispersos en varios dominios administrativos. Es una arquitectura (Fig. 3.1) que, básicamente, está compuesta por módulos basado en *MADS (Middleware Access Drivers)* para conectarse con los diferentes servicios Grid: el módulo para la gestión de ejecución que se comunica a través de sus MADs con los servicios de ejecución Grid, el módulo gestor de información que mediante su MAD se comunica con los Servicios de Información Grid y el módulo gestor de transferencias que mediante su MAD se comunica con los servicios de datos de Grid. GridWay además separa el proceso de planificación del administrador de asignación utilizando un módulo un planificador externo.

En la arquitectura de planificación de GridWay, el planificador es responsable de la asignación de los jobs a los recursos Grid, es decir, que es el que decide cuándo y dónde ejecutar un job, para lo que utiliza la información que proviene de la in-

fraestructura Grid subyacente. Estas decisiones se hacen periódicamente en un bucle infinito aunque la frecuencia de las intervenciones del planificador pueden ajustarse.

La Figura 3.2 describe el procedimiento que realiza el planificador para asignar los jobs a los recursos Grid, recibiendo información de las siguientes fuentes:

1. *Lista de jobs en el sistema*: que incluye jobs pendientes y jobs en ejecución. Los jobs que no se pueden iniciar son filtrados de la lista, es decir, jobs con dependencias insatisfechas, detenidos, etc.

2. *Resultados de Match-making*: los drivers gestores de información (*Information Manager MADs*) consultan los Servicios de Información Grid para realizar un seguimiento de la disponibilidad y el estado de los recursos Grid. El núcleo de GridWay utiliza esta información para construir una lista de recursos adecuados para cada job, es decir, recursos que cumplan los requisitos de esas tareas, y para calcular su rango.

3. *Comportamiento actual del recurso*: el planificador analiza la forma en que un recurso se está comportando en el momento de tomar decisiones. En particular, evalúa la migración y las tasas de fallos, y la ejecución de estadísticas (transferencia, tiempos de ejecución y tiempos de espera en cola).

4. *Utilización pasada del recurso Grid*: el planificador durante el proceso de evaluación también tiene en cuenta el comportamiento que tuvieron los recursos Grid en anteriores ejecuciones.

Finalmente, la información obtenida de las fuentes anteriores se combina con determinadas políticas de planificación que dan prioridad a los jobs y a los recursos. Luego, el planificador envía el job de más alta prioridad al mejor recurso seleccionado para él. El proceso continúa hasta que todos los trabajos se hayan enviado, y los que no se pudieron asignar esperan hasta el próximo intervalo de planificación.

Por lo tanto, se puede observar que si GridWay accede a servicios Índices de MDS4 que concentren un gran volumen de información que se actualiza frecuentemente, el algoritmo de planificación de ejecución de jobs ofrecerá resultados más precisos al utilizar esta información en el proceso de asignación de jobs a los recursos.

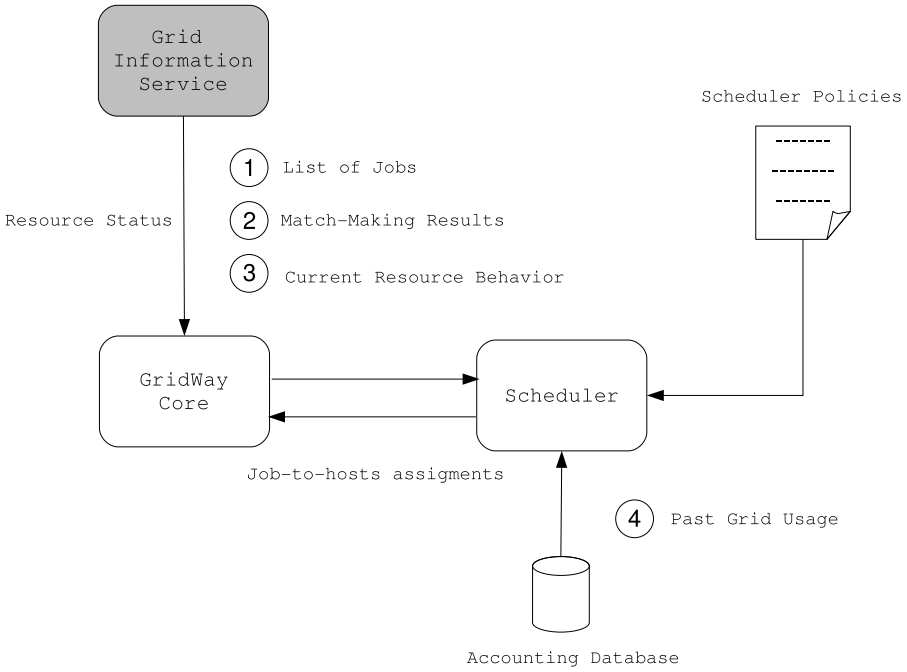

Figura 3.2: Planificación de Jobs en GridWay [12].

Capítulo 4

Multicast IPv6

4.1. Conceptos Básicos

4.1.1. Unicast

En una comunicación unicast existe una fuente y un destino; la relación es uno-a-uno. En este tipo de comunicación las direcciones de la fuente y destino, en un datagrama IP, son direcciones unicast asignadas a las interfaces de esos hosts. En la Figura 4.1, un paquete unicast se envía desde la fuente F1 y pasa a través de los encaminadores hasta alcanzar el destino D1.

Cuando un encaminador recibe un paquete, lo reenvía *sólo a través de una de sus interfaces* (aquella que pertenece al camino óptimo) como está definido en la tabla de encaminamiento. Los encaminadores pueden descartar el paquete si no encuentran la dirección destino en su tabla de encaminamiento.

4.1.2. Multicast

En la comunicación multicast existe una fuente y un grupo de destinos y la relación es uno-a-muchos. En este tipo de comunicación, la dirección de la fuente es una dirección unicast, pero la dirección de destino es una dirección de grupo, un grupo de uno o más destinos/receptores. La dirección de grupo define a los miembros del grupo, que pueden estar localizados en cualquier sitio en internet o en una red privada. La Figura 4.2 muestra este esquema de transmisión de datos.

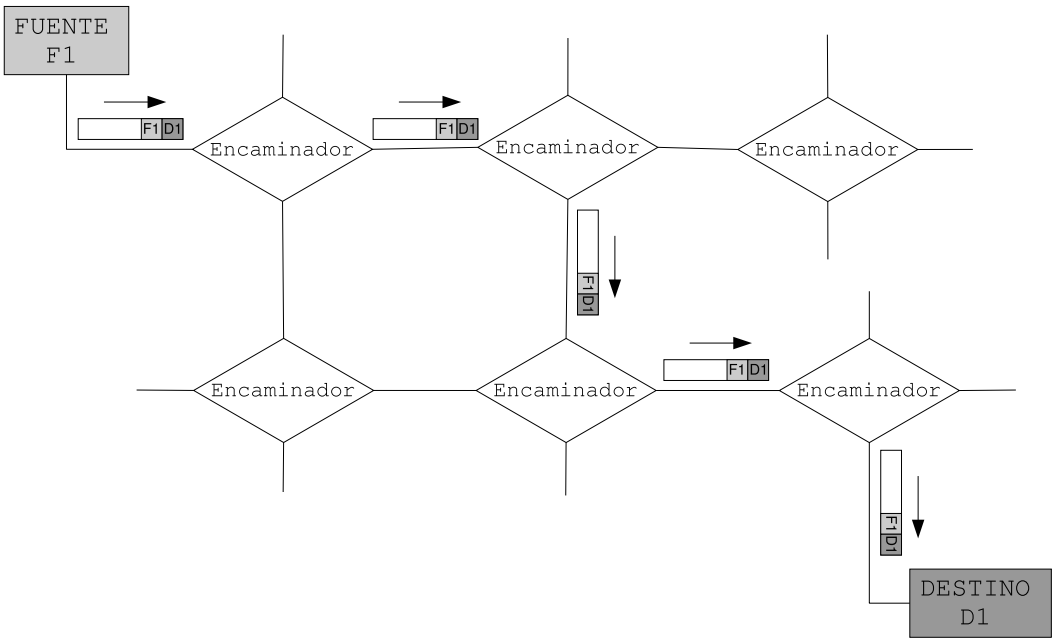

Figura 4.1: Comunicación Unicast.

Un paquete multicast se envía desde la fuente F1 hacia todas los receptores que pertenecen al grupo G1. En la comunicación multicast, cuando un encaminador recibe un paquete, puede reenviarlo a *través de varias de sus interfaces.*

4.1.3. Anycast

En IPv6 se define este nuevo tipo de comunicación donde la relación es similar a multicast, uno-a-muchos, existe una fuente y un grupo de destinos. Sin embargo, un paquete enviado a una dirección anycast se entrega a una de las interfaces identificadas por esa dirección (la más cercana, de acuerdo a la medida de distancia del protocolo de encaminamiento).

4.1.4. Multicast vs Unicast Múltiple

La comunicación multicast envía un único paquete desde una fuente, este paquete es duplicado por los encaminadores. La dirección de destino en cada paquete es la misma para todos los paquetes duplicados. Una sóla copia del paquete se envía entre encaminadores.

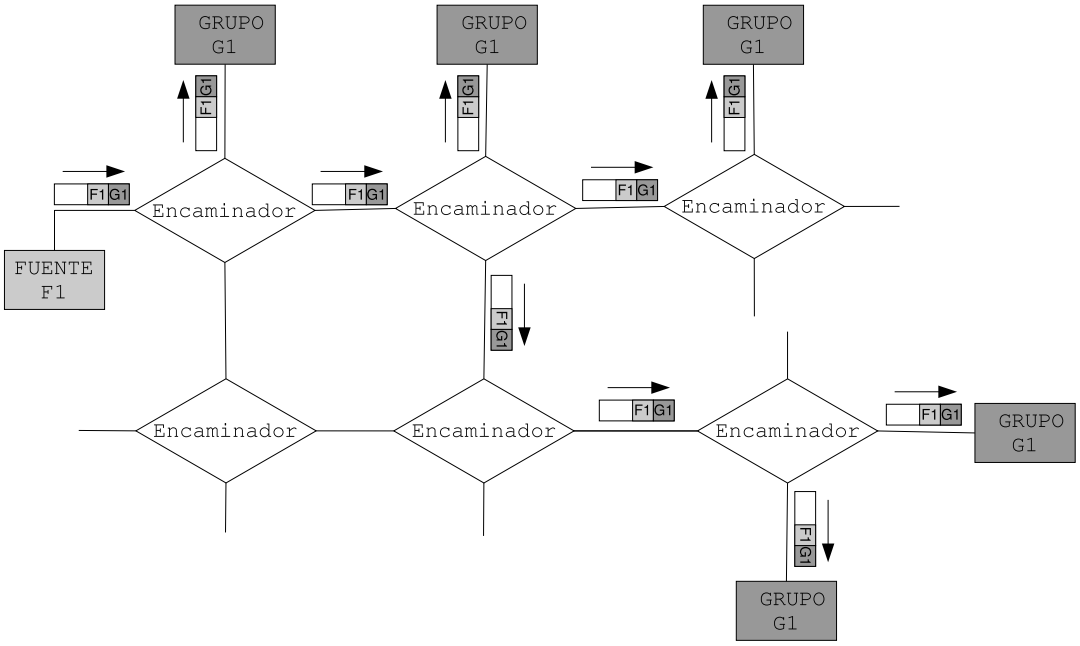

Figura 4.2: Comunicación Multicast.

En la comunicación unicast múltiple, la fuente envía varios copias del paquete, cada uno con una dirección unicast de destino diferente. De esta forma pueden existir múltiples copias transmitiéndose entre encaminadores.

La comunicación multicast es más eficiente que la simulada con unicast por dos razones:

- El multicast requiere menos ancho de banda que la transmisión unicast múltiple donde algunos enlaces entre encaminadores pueden tener que transmitir varias copias del paquete.
- En una transmisión unicast múltiple, la generación de los paquetes en la fuente puede presentar demoras y en grupos grandes esta demora entre el primero y el último paquete puede ser inaceptable. En multicast no hay demora, porque la fuente crea y envía solamente un paquete.

4.1.5. Aplicaciones de Multicast

La comunicación multicast puede aplicarse en diversos situaciones:

- Acceso a base de datos distribuidas.

- Distribución de software y de información.
- Servicios de tiempo.
- Servicios de nombre, como DNS.
- Replicación de base de datos.
- Video y audio streaming.
- Servicios de descubrimiento.
- Computación distribuida.
- Educación a distancia.

En estas aplicaciones basadas en multicast, las transmisiones de datos entre la fuente y el/los destino/s implican tanto la identificación de las direcciones origen y destino así como un esquema de encaminamiento que optimice la entrega de los datos desde la fuente hacia el destino. Por tanto, los conceptos claves en las comunicaciones multicast incluyen una dirección IP de grupo multicast, un árbol de distribución y los receptores interesados en recibir información enviada a esa dirección de grupo.

4.2. Registro en el Grupo Multicast

En [40] se describen qué pasos deben seguirse para establecer una comunicación multicast entre nodos de una red. El primero de ellos es la identificación de los receptores, conocido como *proceso de registro*, que permite a los receptores unirse a grupos multicast existentes. El encaminador local usará este proceso para determinar dónde deberá enviar los datos multicast.

IPv6 tiene su propio protocolo de registro de receptores llamado *Multicast Listener Discovery (MLD)*. MLDv2 [44] es un subprotocolo de ICMPv6 [8], esto significa que los tipos de mensajes de MLDv2 son un subconjunto de los mensajes ICMPv6. Los encaminadores IPv6 utilizan MLDv2 para descubrir la presencia de oyentes multicast (los nodos que desean recibir paquetes multicast) en los enlaces conectados directamente a él y para descubrir específicamente qué direcciones multicast son de interés para esos nodos vecinos. El encaminador multicast puede ser en sí mismo un oyente de una o más direcciones multicast; en este caso se desempeña tanto en el *rol de encaminador multicast* para recoger información de oyentes multicast necesaria para su protocolo de encaminamiento multicast como en el *rol de oyente de dirección*

multicast para informarse a sí mismo y a otros encaminadores multicast vecinos de su estado de oyente.

Por lo tanto, MLD es un protocolo asimétrico, es decir que especifica comportamientos separados para oyentes de direcciones multicast (hosts o encaminadores) y para encaminadores multicast. El propósito de MLD es permitir a cada encaminador multicast aprender, para cada uno de los enlaces conectados directamente a él qué direcciones multicast y qué fuentes tienen oyentes interesados en ese enlace. La información recogida por MLD es suministrada a cualquier protocolo de encaminamiento multicast usado por el encaminador, para asegurar que los paquetes multicast sean entregados a todos los enlaces donde hay oyentes interesados en tales paquetes.

4.3. Encaminamiento multicast

El encaminador que realiza el proceso de registro de receptores multicast, como se explicó anteriormente, se denomina *Last Hop Router (LHR)* ya que es contactado por los receptores y se encuentra en el final de la trayectoria de comunicación. Contrariamente, *el First Hop Router(FHR)* es el encaminador más cercano a la fuente del tráfico multicast.

Se requiere un mecanismo de control inteligente para que los datos puedan ser eficientemente distribuidos desde el *First Hop Router* hacia todos los *Last Hop Routers* participantes. Esto se implementa a través de un protocolo de encaminamiento. Las funciones [41] que se requieren para llevar a cabo el proceso de encaminamiento de multicast son las siguientes:

1. Establecer una convención para identificar direcciones multicast: en IPv6 la dirección consta de un prefijo de 8 bits con valor 1 más un campo de indicadores de 4 bits (indicando si la red está permanentemente asignada o no), un campo de ámbito de 4 bits (ámbito de aplicabilidad de la dirección, rango que abarca desde una única red a la red global) y el identificador de grupo de 112 bits.

2. Cada encaminador debe traducir una dirección multicast IPv6 a una lista de redes que contengan miembros del grupo multicast. Esta información le permite construir un árbol de distribución de camino más corto (*shortest path tree, SPT*) hacia todas las redes que contengan miembros del grupo.

3. El encaminador debe traducir una dirección IP multicast a una dirección de red para poder entregar en la red destino el datagrama multicast IPv6. (Por

ejemplo, en redes IEEE 802, una dirección de nivel MAC es de 48 bits. Si el bit más significativo es 1, entonces es una dirección multicast, un encaminador conectado a una red IEEE 802 debe traducir una dirección multicast IPv6 de 128 bits a una dirección multicast de nivel MAC de IEEE 802 de 48 bits).

4. Las direcciones multicast pueden ser permanentes, pero usualmente se generan de forma dinámica y los hosts pueden unirse o abandonar los grupos de multicast dinámicamente. De esta manera, se necesita un mecanismo mediante el cual un host individual informe al encaminador conectado a su misma red de su inclusión o exclusión en un grupo multicast, tal como se describió en el proceso de registro multicast en la Sección 4.2.

5. Por último, los encaminadores deben intercambiar dos tipos de información:

   - Qué redes contienen miembros de un grupo multicast particular.
   - Información para calcular los caminos más cortos a cada red que contenga miembros del grupo, aplicando los protocolos de encaminamiento multicast.

6. Se necesita un algoritmo de encaminamiento para calcular los caminos más cortos a todos los miembros del grupo.

7. Cada encaminador debe determinar la ruta de distribución multicast basándose en las direcciones fuente y destino.

Si en el esquema de encaminamiento unicast, cada encaminador en el dominio tiene una tabla que define el árbol de camino más corto a los posibles destinos, por el contrario, cuando un encaminador recibe un paquete multicast, este puede tener que ser enviado a más de un destino en más de un red. Por tanto, el reenvío de un único paquete a los miembros de un grupo requiere un árbol SPT. Si existen n grupos, el encaminador precisa n SPT. Se desarrollaron dos enfoques para resolver el problema: los árboles *source-based* y los árboles *group-shared*. De esta forma, en el encaminamiento multicast, cada encaminador involucrado necesita construir un SPT basándose en algunos de estos enfoques:

- Árbol *source-based*: cada encaminador precisa tener un SPT para cada grupo. Este SPT define el próximo salto para cada red que tenga miembros registrados para ese grupo. Si el número de grupos es m, cada encaminador precisa tener m SPT, uno para cada grupo. Por tanto, a medida que aumenta la cantidad de grupos también lo hace el tamaño de la tabla de encaminamiento.

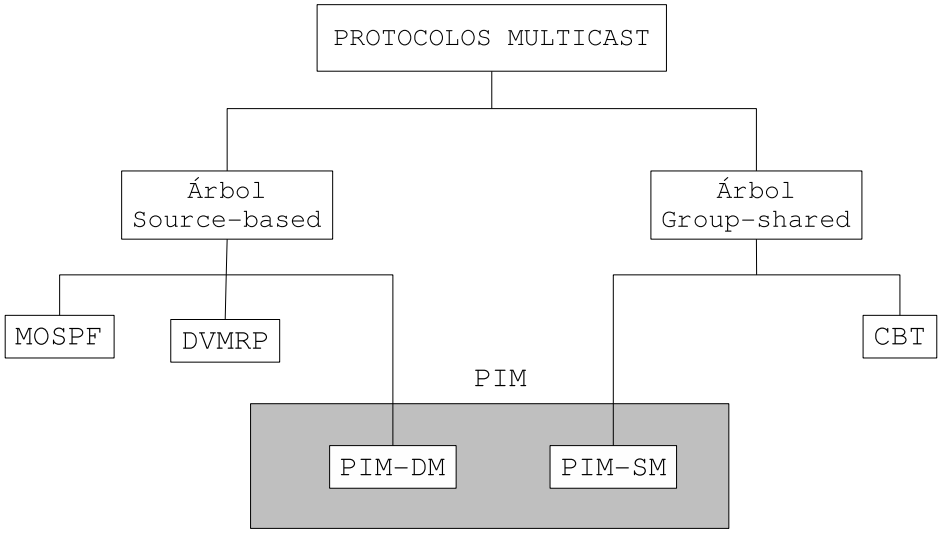

Figura 4.3: Clasificación de Protocolos Multicast.

- Árbol *Group-shared*: en lugar de que cada encaminador tenga m SPT, solamente un encaminador designado, llamado *Rendevouz Point (RP)*, se encarga de la distribución del tráfico multicast. El RP tiene m SPT en su tabla de encaminamiento. El resto de los encaminadores en el dominio no tienen ninguno. Si un encaminador recibe un paquete multicast, lo encapsula en un paquete unicast y envía al RP. El RP desencapsula el paquete multicast y consulta su tabla de encaminamiento para dirigir el paquete. Es decir que, en este enfoque, sólo el RP, que tiene un SPT por cada grupo, está involucrado en la transmisión multicast.

4.3.1. Protocolos de Encaminamiento

Los protocolos de encaminamiento multicast, puede clasificarse [15] como se muestra en la Figura 4.3.

DVMRP [45] (*Distance Vector Multicast Routing Protocol*) es el más antiguo de los protocolos para encaminamiento multicast. Un concepto clave introducido por DVMRP fue el uso de distintos árboles de reenvío para cada uno de los grupos multicast; este principio fundamental se continúa usando en los nuevos protocolos de encaminamiento multicast. DVMRP es análogo al RIP [28], es decir, que es un protocolo por vector de distancia que proporciona una muy limitada flexibilidad, funcionalidad y escalabilidad.

MOSPF [32] (*Multicast Open Shortest Path First*) es una extensión del protocolo OSPF [33], y como tal, debe residir dentro de los límites de un sistema autónomo y requiere la utilización del protocolo OSPF para poder funcionar. En consecuencia, sólo tiene significado dentro de una corporación, universidad u otra organización que dé soporte al encaminamiento multicast pero no puede soportar aplicaciones a gran escala que requieran el uso de Internet.

El resto de los protocolos aportan sus propias soluciones tecnológicas, utilizando diversas técnicas para construir el esquema de árbol de distribución multicast. Para mayor detalle acerca de cada uno de ellos, las especificaciones se pueden consultar PIM-DM [1], CBT [3, 2], PIM-SM [13, 38]. Debido a su importancia, en la próxima Sección se describe la familia de protocolos PIM.

4.4. Protocol Independent Multicast (PIM)

A finales de la década del 90 se desarrolló una nueva especie de protocolos multicast. Esta familia de protocolos es conocida en su conjunto como *Protocol Independent Multicast* (PIM). Son protocolos de encaminamiento multicast que se caracterizan porque no dependen de ningún protocolo de encaminamiento específico sino que aprovechan las tablas de encaminamiento existentes para reenviar datos multicast sin tener en cuenta cómo fueron construidas. Los protocolos PIM-DM (*Protocol Independent Multicast Dense-Mode*) y a PIM-SM (*Protocol Independent Multicast Sparse-Mode*) son los más extendidos y se describen a continuación, aunque existen otras versiones de PIM más recientes como *PIM Bidirectional Mode (Bidir-PIM)* [24] o *PIM Source Specific Mode (PIM-SSM)* [6].

4.4.1. PIM Dense Mode (PIM-DM)

Es un protocolo recomendado cuando existe la posibilidad de que cada encaminador participe de la distribución multicast (*dense mode*) y para entornos multicast numerosos (*dense*), tales como las redes LAN. Es un protocolo que implementa el enfoque de árbol *source-based*, asume que los sistemas autónomos están usando un protocolo unicast y cada encaminador tiene una tabla donde puede encontrar la interfaz de salida que tiene el camino óptimo hacia el destino.

PIM-DM implementa un mecanismo *push*, es decir, *empujar* los datos multicast hacia los receptores. Un encaminador que implemente este protocolo inunda con flujos de tráfico multicast a todas las interfaces. Si los encaminadores *downstream* (los que se encuentran *corriente abajo* conectados a partir de este encaminador)

no tienen conectados ningún receptor que requiera de este flujo de datos multicast en particular, enviarán un mensaje *stop* hacia el encaminador *upstream* (*corriente arriba*, hacia el encaminador que envió el flujo de tráfico multicast). Estos mensajes se conocen como *prune* (poda) ya que el encaminador *upstream* podará su árbol de reenvío para eliminar esa *rama* en particular. Los encaminadores PIM-DM envían el tráfico multicast hasta que un encaminador *downstream* rechaza el flujo.

4.4.2. PIM Sparse Mode (PIM-SM)

Este protocolo es recomendado cuando existe alguna posibilidad de que el encaminador participe en la distribución multicast (*sparse mode*) y es apropiado para entornos multicast *dispersos* como las redes WAN. Es un protocolo que implementa el enfoque de árbol *group-shared*, es decir que tiene un RP como fuente del árbol; pero puede intercambiar esta estrategia con la del árbol *source-based*, por ejemplo en el caso que exista un área numerosa de actividad lejos del RP, ésta puede ser administrada más eficientemente con un estrategia de árbol *source-based* que con una estrategia *group-shared*.

PIM-SM es la implementación más común de PIM sin embargo, contrasta directamente con PIM-DM ya que utiliza un mecanismo de *pull* (*solicitud*) en lugar de una técnica *push* (*empuje*). Esto significa que los encaminadores PIM-SM deben solicitar específicamente un flujo multicast particular antes de que los datos sean reenviados hacia ellos. PIM-SM se adapta a internet ya que reduce la sobrecarga y ancho de banda para flujos de datos multicast.

Es un protocolo que existe exclusivamente entre encaminadores, es decir que los hosts tanto fuentes como receptores no participan en este protocolo. Los encaminadores PIM-SM generan periódicamente un mensaje *Hello* para descubrir y mantener sesiones de estado con los vecinos. Después de que los vecinos son descubiertos, los encaminadores PIM-SM pueden comunicar su interés en unirse a grupos multicast específicos. Esto se lleva a cabo por medio del envío de un mensaje PIM-SM *join* explícito desde un encaminador *downstream* hacia un encaminador *upstream*. El mensaje *join* especifica el grupo y la fuente a los que el encaminador quiere unirse. Los encaminadores *upstream* pueden entonces reenviar información multicast a los encaminadores *downstream*. Este comportamiento es completamente opuesto al de PIM-DM, el cual envía la información a todos los encaminadores hasta que éstos le indican que deje de hacerlo.

4.5. Observaciones sobre el Encaminamiento Multicast

Es importante tener en cuenta las siguientes observaciones relacionadas con el encaminamiento multicast IPv6:

- Para establecer comunicaciones multicast de manera satisfactoria el primer paso consiste en la identificación de los receptores utilizando el protocolo de la capa 3 del modelo TCP/IP para IPv6, MLDv2. Una vez que los receptores se han unido a sus respectivos grupos, la red debe distribuir el tráfico multicast a los destinos finales correctos, proceso que es llevado a cabo mediante un protocolo de encaminamiento multicast, siendo la familia de protocolos PIM la más extendida para permitir el encaminamiento multicast de los paquetes de datos.
- MRD6 [37] es un software que implementa un encaminador multicast para Linux con soporte para los principales protocolos de encaminamiento multicast como MLDV2 y PIM-SM. Es un solución software muy reciente que es escalable y cuenta con soporte para IPv6 nativo o virtual (a través de túneles).
- Los flujos de datos multicast no soportan protocolos fiables de capas superiores, tales como TCP, debido a que la fuente de datos no conoce cuántos hosts *downstream* están recibiendo sus datos y por lo tanto es imposible mantener conexiones TCP fiables con todos los usuarios finales, en su lugar, comúnmente, se utiliza UDP para distribuir el tráfico multicast.
- Multicast tiene soporte por parte de los encaminadores sí y sólo sí éstos son explícitamente habilitados y configurados por sus administradores. Actualmente algunos proveedores de servicios de Internet dan soporte para la transmisión multicast IPv4 a través de sus encaminadores, pero no todos ellos, pues las redes IPv6 nativas aún no están tan extendidas a nivel mundial como las redes IPv4.

4.6. Implementación de Multicast IP

El esquema de transmisión multicast IP, según la pila de protocolos TCP/IP (Fig. 4.4), se implementa en la capa de transporte a través del Protocolo de Datagrama de Usuario (UDP). A continuación se explican brevemente sus principales características.

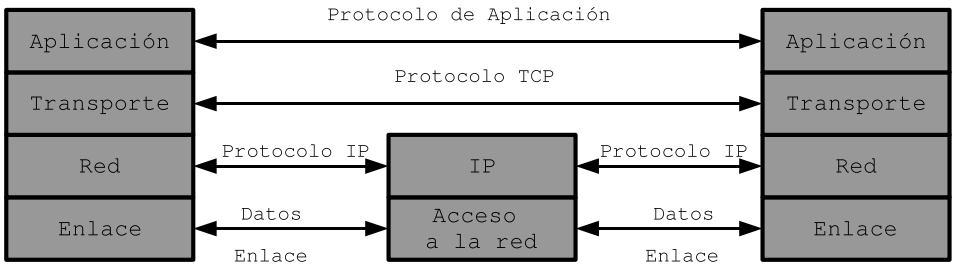

Figura 4.4: Pila de Protocolos TCP/IP.

UDP es un protocolo no orientado a conexión, es decir, que no garantiza la entrega de los datos al destino transmitidos en lo que se conoce como *datagrama UDP*. Se define un datagrama como un transmisión única que puede ser entregada cero o más veces. Su secuencia con respecto a otros datagramas entre los extremos finales de la comunicación no esta garantizada, por lo tanto, pueden ser enviados desordenadamente, no ser enviados o ser enviados varias veces. Cada datagrama se envía en un único paquete IP pero está sujeto a restricciones de tamaño:

- El protocolo IPv4 los limita a 65507 bytes.
- Los encaminadores IPv4 están autorizados a fragmentar cualquier paquete IP, incluyendo segmentos TCP y datagramas UDP. Pero, al contrario de lo que ocurre con los segmentos TCP, un datagrama UDP no puede ser reconstruido una vez que ha sido fragmentado y por lo tanto se descarta.
- IPv6 define los llamados *jumbogramas* en el nivel IP; esto permite datagramas UDP de hasta $2^{32} - 1$ bytes [7].

Los beneficios que aporta UDP se resumen en los siguientes:

- UDP, a diferencia de TCP, es un protocolo que al no estar orientado a conexión no produce sobrecarga en la red, pues no existe comunicación previa para realizar la conexión ni la desconexión. Para establecer la comunicación, UDP sólo utiliza la información de direccionamiento que el propio datagrama incorpora en su cabecera.
- La arquitectura de los servidores UDP es mucho más simple que la arquitectura de los servidores TCP, ya que no existen sockets de aceptación ni de cierre.
- La arquitectura de los clientes UDP es también algo más simple que la de los clientes TCP, al no existir creación ni finalización de conexión.

Las limitaciones de UDP pueden resumirse como sigue:

- No hay soporte para el reensamblado de datagramas fragmentados.
- No hay soporte para secuenciamiento de paquetes.
- No hay detección de paquetes perdidos y retransmisiones.

A pesar de estas limitaciones, el modelo de datagramas se ajusta perfectamente para aplicaciones en las que:

- Las transacciones son de tipo solicitud-respuesta.
- La carga útil es pequeña.
- Los servidores son *stateless*, es decir, que el servidor trata cada solicitud como una transacción de forma única, totalmente independiente de cualquier otra solicitud anterior.
- Las transacciones son *idempotentes*, es decir, que pueden ser repetidas sin afectar el resultado total del cómputo o proceso.

4.7. Beneficios y Limitaciones de Multicast

Los beneficios de multicast, comparando la transmisión multicast con respecto a la transmisión *unicast múltiple* de la misma información a los mismos destinatarios, se resumen a continuación:

- Existe un considerable ahorro en ancho de banda de red y, por tanto, del uso de la red en términos económicos: si el costo de transmitir usando unicast una determinada cantidad de datos a N destinatarios es de \$1, el costo de transmitir con multicast los mismos datos a los mismos destinatarios es de $\$1/N$. También hay un ahorro en el tiempo de propagación de los datos a los destinatarios, ya que el total de transmisión se completa en $1/N$ del tiempo en comparación con la transmisión unicast. Con la ventaja secundaria de la reducción de la carga procesamiento en el servidor.
- Otro beneficio de multicast es que todos los destinatarios reciben los datos al mismo tiempo, si no se considera la pérdida de paquetes y la retransmisión. Este beneficio puede ser aprovechado por aplicaciones *time-sensitive* tales como la distribución de cotizaciones bursátiles o, como se verá en los siguientes

Capítulos, por aplicaciones que dan soporte a Servicios de Información en entornos Grid.

- En comparación con unicast, multicast es eficiente en términos económicos a medida que aumenta la cantidad de destinatarios. Se puede decir que multicast es una solución que escala mucho mejor que unicast.
- Cuando el número de destinos es muy grande, aplicaciones a gran escala, como la transmisión de video a través de Internet son técnica y económicamente viables, ya que nunca podrían ser factibles a través de unicast.

Las limitaciones de multicast principalmente son las siguientes:

- Al tratarse de un mecanismo UDP no ofrece garantías en la entrega de los datos.
- Requiere la cooperación/soporte de los encaminadores para poder funcionar.
- La seguridad en las transmisiones multicast son muy complejas, excepto en el caso que la red subyacente sea IPv6, donde la seguridad se implementa nativamente en la capa de red.

Capítulo 5

Trabajos Relacionados

En entornos Grid, además de los Servicios de Información de GT4, existen ciertas aplicaciones que pueden beneficiarse de la transmisión multicast. Es el caso de aplicaciones que requieren la distribución de terabytes de datos que generan las mediciones de instrumentos científicos desde una ubicación determinada hacia varios recursos de almacenamiento o procesamiento ubicados en otros sitios para ser procesados [27]. Otras aplicaciones donde se podría aplicar multicast son aquellas donde, por ejemplo, simulaciones a gran escala generan cientos de terabytes de datos y requieren el envío de esa información desde el sitio donde se originan hacia otros sitios remotos [5].

La implementación de estas aplicaciones Grid, a su vez, también produce sobrecarga en el tráfico en la red. Por lo tanto, para disminuir la utilización del ancho de banda de la red y debido al esquema de distribución de datos de estas aplicaciones, es que se han desarrollado diversos trabajos donde se propone la transmisión multicast para optimizar el funcionamiento de dichas aplicaciones.

Por ejemplo, en [26] se estudia la utilización de multicast en el nivel de capa de transporte sobre TCP, a través del desarrollo del protocolo TCP-XM. Este protocolo permite la transferencia multicast de datos de manera fiable hacia un grupo de receptores por medio de una solución que combina la transmisión TCP unicast y la transmisión multicast simultáneamente. Es una propuesta que sólo fue probado de manera experimental en Globus Toolkit 3 y aplicada a la transferencia de grandes volúmenes de datos.

Por otra parte, existen trabajos que han aplicado la transmisión multicast en el nivel de capa de aplicación tales como [4, 46, 31]. Estos trabajos, al igual que

el anterior, tienen como objetivo mejorar los servicios de gestión de datos como la transferencia de datos.

Sin embargo, las propuestas para aplicar la transmisión multicast a los Servicios de Información de los sistemas Grid son escasas. En particular, en [30] se estudió el descubrimiento de recursos pero aplicado a sistemas Grid ad-hoc, los que se caracterizan por estar basados en una infraestructura de red que no está fija, donde los participantes de las VOs pueden unirse y abandonar las mismas sin previo aviso, y en consecuencia, los recursos y servicios también. La solución que propone este trabajo es un mecanismo de descubrimiento de recursos híbrido basado en el descubrimiento de zona y donde se aplica la transmisión multicast nativa sobre IPv4.

Capítulo 6

Modelo de Organización de Servicios Índices Basado en Multicast IPv6

Se diseñó un esquema de organización de servicios Índices de MDS4 comunicados mediante multicast IPv6 (Fig. 6.1). Los elementos básicos, comunes a los modelos que se verán en las siguientes secciones:

- Las fuentes emisoras de tráfico multicast corresponden a los servicios Índices de MDS4 ubicados en los nodos de mayor jerarquía en cada sitio de una VO.
- Cada fuente emisora de tráfico multicast recibe el nombre de *nodo recolector mcast*.
- El grupo multicast está integrado por aquellos servicios Índices de MDS4 que serán consultados por GridWay, cada uno de ellos se denomina servicio *Index Mcast*, para reflejar su pertenencia al grupo multicast.
- Cada receptor integrante del grupo multicast recibe el nombre de *nodo receptor mcast*.
- La solución software que implementa transmisión multicast para enviar la información registrada en los servicios Índices residentes en los nodos recolectores mcast, recibe el nombre de *agente recolector mcast*.
- La solución software que recibe y procesa la información enviada al grupo multicast se encuentra instalada en cada nodo receptor mcast y se denomina *agente receptor mcast*.

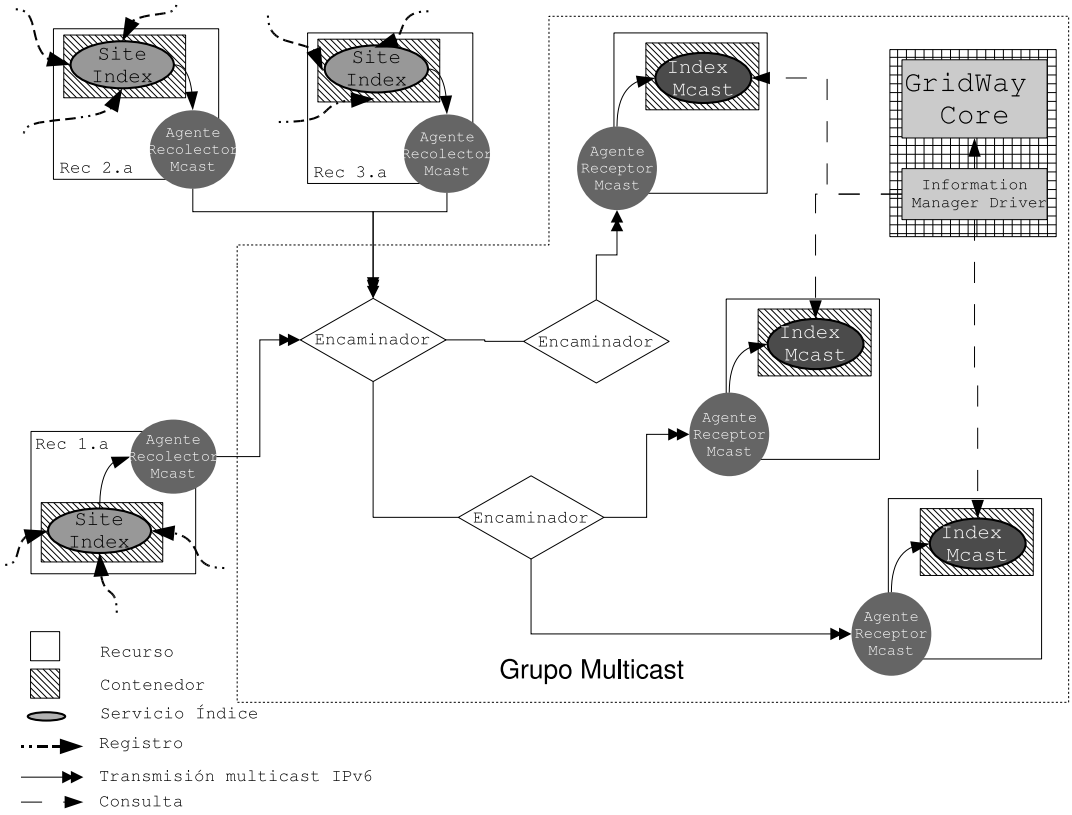

Figura 6.1: Servicios Índices MDS4 comunicados mediante multicast IPv6.

Por encima de la organización jerárquica de servicios Índices MDS4 en cada sitio de una VO, los servicios *Index Mcast* concentran información proveniente de esos servicios Índices que enviaron sus datos mediante multicast IPv6 a través de sus respectivos *agentes recolectores mcast*. Los servicios Índices de mayor nivel jerárquico no requieren registrarse en los servicios *Index Mcast*.

El metaplanificador GridWay puede configurarse para consultar a uno o a varios servicios *Index Mcast*. GridWay accede a los servicios *Index Mcast* en el momento de realizar la planificación Grid según el proceso que se describió en el Capítulo 3.

La información acerca de los servicios GRAM registrados en los servicios Índices de cada *nodo recolector mcast*, se transmite mediante multicast IPv6 al grupo multicast integrado por los servicios *Index Mcast*. El esquema resultante de servicios Índices se caracteriza por constituir una estructura plana, redundante y tolerante a fallas que puede ser consultado por GridWay.

Según lo expuesto en la Sección 2.3, se diseñó un *proveedor de información* que, a través de una *fuente agregadora de ejecución* y mediante multicast IPv6, proporciona esa información a los servicios *Index Mcast*.

En las siguientes secciones se describen detalladamente los diseños de modelos de organización de servicios Índices de MDS4 basados en multicast IPv6 con y sin soporte para la transmisión segura de los datos.

6.1. Modelo No Seguro de Organización de Servicios Índices basado en Multicast IPv6

La Figura 6.2, muestra de qué forma el proveedor de información utiliza multicast IPv6 para enviar los datos referentes a los servicios GRAM obtenidos del servicio Índice en cada nodo recolector mcast. En cada nodo receptor mcast, el proveedor de información proporciona la información con que la fuente agregadora de ejecución alimentará al servicio *Index Mcast*. Las funciones de los agentes según este modelo son las siguientes:

- *Agente Recolector Mcast*: A través de un usuario válido de GT4 consulta los servicios GRAM registrados en el servicio Índice del nodo donde el agente se ejecuta. Luego, por cada servicio GRAM, genera un fichero que comprime y transmite al grupo multicast.
- *Agente Receptor Mcast*: Procesa cada datagrama entrante sólo si la información contenida no está obsoleta con respecto a un tiempo de validez de la información predeterminado.

6.2. Necesidades de Seguridad del Modelo Propuesto

GSI es un componente de GT4 que provee servicios de seguridad básicos necesarios para dar soporte a los sistema Grid. Foster, en [16], lo define como un conjunto de componentes de GT4 basados en estándares de seguridad que implementan la seguridad utilizando credenciales y protocolos para la protección del mensaje, autenticación, delegación y autorización. Como se muestra en la Figura 6.3, GSI da soporte para (a) seguridad a nivel de mensajes compatible con *WS-Security* y credenciales X.509 (lento), (b) soporte para nombres de usuario y contraseñas (inseguro, pero compatible con *WS-I Base Security Profile*) y para (c) seguridad a nivel de transporte con credenciales X.509 (más rápido y, por tanto, el valor por defecto).

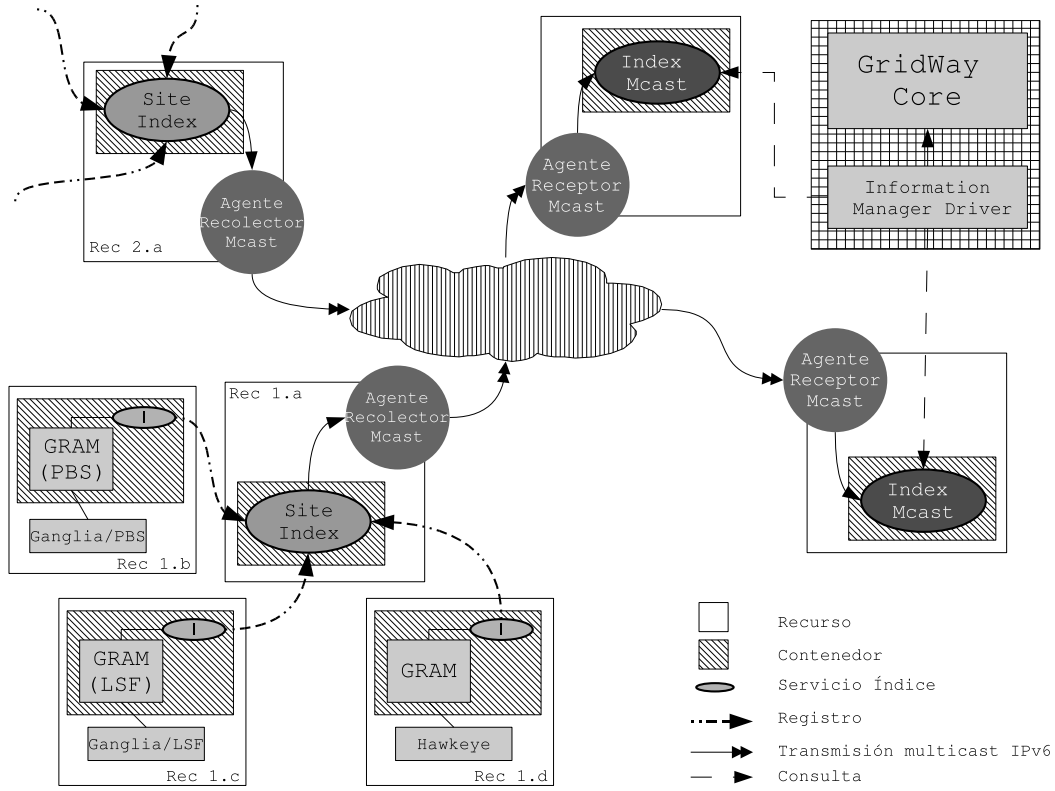

Figura 6.2: Modelo No Seguro de Organización de servicios Índices basado en multicast IPv6.

En la configuración de GT4 por defecto, cada usuario y recurso cuenta con un certificado X.509. Los protocolos se implementan para permitir que dos entidades validen las credenciales mutuamente, para utilizar esas credenciales en el establecimiento de un canal seguro con el propósito de proteger los mensajes, y para crear y transportar credenciales delegadas que permiten a un componente remoto actuar en nombre de usuario por un período limitado de tiempo [21, 20].

Por defecto, GSI no establece comunicación confidencial entre las partes, es decir, no hay encriptación de la información. Una vez que se produjo la autenticación mutua de las partes, GSI permite que la comunicación puede efectuarse sin la sobrecarga de la constante encriptación y desencriptación. No obstante, si provee por defecto integridad de comunicación.

Considerando lo anterior, y como veremos en la siguiente Sección, el modelo seguro de organización de servicios Índices, tampoco considera la encriptación de datos, pues no es un modelo que esté centrado en mantener la privacidad de los datos, sino sólo en que dichos datos no se hayan modificados y que el origen que envía la información sea auténtico.

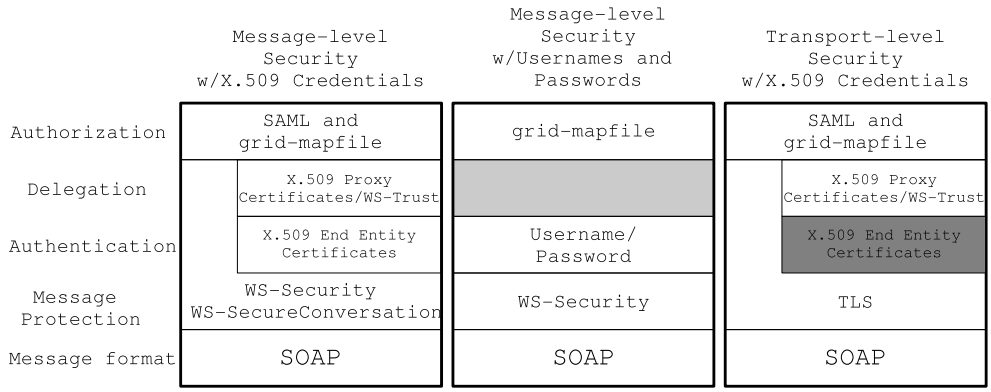

Figura 6.3: Protocolos de Seguridad de GT4 [47].

Por otra parte, los autores de [14], consideran las implicaciones de seguridad en las comunicaciones multicast IP, independientemente de la versión, basándose en las diferencias entre la transmisión unicast y multicast.

- La comunicación unicast consiste en una conversación explícita entre dos participantes. Por lo tanto, tiene sentido que la seguridad se base en estos participantes. Asimismo, la *confianza* puede basarse en la confianza en cada participante y en la confianza en los datos.
- La comunicación multicast, por el contrario, implica un tamaño arbitrario, potencialmente variable de un conjunto de participantes, cuyo número probablemente no pueda determinarse. Debido a esto, la seguridad en la comunicaciones multicast no se basa en la *confianza* de sus participantes sino más bien en la confianza de sus *datos*. En particular, la comunicaciones multicast son autenticadas mediante la autenticación de paquetes de datos, por ejemplo, utilizando la firma digital y privacidad que se obtiene mediante la encriptación de estos datos.

De esta manera, teniendo en cuenta las vulnerabilidades de la transmisión multicast y las herramientas de seguridad, que a su vez, proporciona GT4 mediante GSI, es posible y necesario que el modelo propuesto incorpore seguridad tanto para autenticar la fuente emisora como para proteger la integridad de los datos transmitidos.

6.3. Modelo Seguro de Organización de Servicios Índices basado en Multicast IPv6

En este diseño de organización de servicios Índices los datos transmitidos al grupo multicast se envían luego de haber sido firmados digitalmente en cada nodo recolector mcast. En el nodo receptor mcast los datos son verificados y autenticados antes de alimentar a cada servicio *Index Mcast*.

La Figura 6.4, muestra de qué forma el proveedor de información entrega mediante multicast IPv6 y de manera segura los datos referentes a los servicios GRAM obtenidos del servicio Índice en cada nodo recolector mcast. En el nodo receptor mcast el proveedor de información verifica la integridad y autenticidad antes de proporcionar esa información a la fuente agregadora de ejecución para alimentar al servicio *Index Mcast*. Las funciones de los agentes según este modelo son las siguientes:

- *Agente Recolector Mcast*: A través de un usuario válido de GT4, el agente consulta los servicios GRAM registrados en el servicio Índice del nodo donde el agente se ejecuta. Luego, por cada servicio GRAM, genera un fichero que firma digitalmente, empaqueta, comprime y transmite al grupo multicast.
- *Agente Receptor Mcast*: Procesa cada datagrama entrante sólo si verifica su integridad y autenticidad y si la información contenida en él no está obsoleta con respecto al tiempo de validez de información predeterminado. Finalmente procesa cada datagrama entrante sólo si la información contenida no está obsoleta con respecto a un tiempo de validez de la información predeterminado.

Para proteger los datos este nuevo diseño de modelo de organización de servicios Índices basado en multicast IPv6, utiliza parte de los componentes de GSI. Se aplica criptografía de clave pública [29] para autenticar las fuentes que envían los datos al grupo multicast de servicios *Index Mcast*. Los certificados digitales previenen los ataques de tipo *man-in-the-middle* en el caso que un intruso puede suplantar a los nodos recolectores mcast.

En nuestro modelo, la transmisión de información sobre multicast IPv6 es protegida mediante la verificación de la integridad y autenticidad de esa información. Como cada usuario en GT4 cuenta con un certificado X.509, se utiliza esa credencial para realizar la validación de la autenticidad de la fuente que envía los datos al grupo multicast y la verificación de la integridad de dichos datos. Como se observa en la Figura 6.4, el agente recolector mcast genera un resumen usando un algoritmo hash

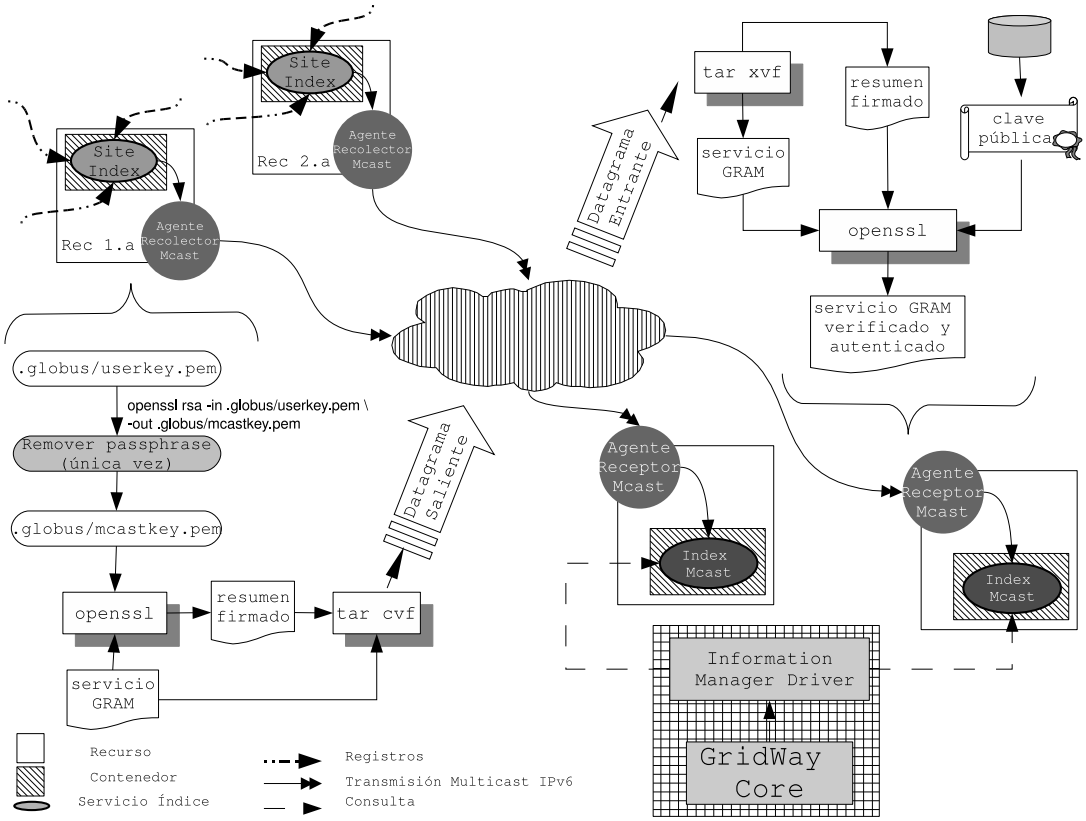

Figura 6.4: Modelo Seguro de Organización de servicios Índices basado en multicast IPv6.

(SHA1) por cada fichero correspondiente a un servicio GRAM y lo firma usando la biblioteca OpenSSL [36] mientras que el agente receptor mcast verifica la integridad de los datos y autenticidad usando la clave pública del nodo recolector mcast y, en caso de ser válida, procesa los datos para alimentar al servicio *Index Mcast*.

6.4. Detalles de Funcionamiento del Modelo Seguro

6.4.1. Proceso de Firmas

Para firmar cada pieza de información, el agente recolector mcast primero calcula el hash matemático de la información (una versión condensada de la información), SHA1 en este caso, dicho algoritmo también es conocido por los receptores de la información. Con la clave privada del usuario válido de GT4, el agente recolector mcast encripta el hash y lo adjunta al mensaje, es decir, al fichero correspondiente

a cada servicio GRAM. Luego comprime esos ficheros y los envía a la dirección de grupo multicast.

Una vez que el nodo receptor mcast recibe el datagrama, el correspondiente agente verifica primero la autenticidad del emisor, o sea, la validez del certificado del emisor utilizando los certificados almacenados en el directorio de CAs de confianza de GT4. Si la identidad del usuario que envió los datos se autenticó correctamente, el agente verifica a continuación la integridad del mensaje, para ello calculará el hash del mensaje usando el mismo algoritmo que utilizó el agente recolector mcast y luego desencriptará el hash encriptado que recibió junto con el mensaje. Si el nuevo hash calculado y el hash desencriptado coinciden, entonces se comprueba que el agente recolector mcast firmó el mensaje y que el mensaje no fue modificado. De esta manera el agente receptor mcast recién procesará los datos para alimentar al servicio *Index Mcast*, verificando previamente si los mismos no se encuentran obsoletos con respecto al tiempo de validez de información predeterminado.

6.4.2. Distribución de Certificados

Antes de enviar los datos correspondientes a los servicios GRAM registrados en el MDS4 de ese nodo, cada nodo recolector mcast debe distribuir al grupo multicast el certificado digital asociado a un usuario de GT4 válido. La distribución se realiza por defecto con una frecuencia de 5 minutos, pero el valor puede ser ajustado por el administrador de cada nodo recolector mcast. De este modo se logra, principalmente, disminuir la sobrecarga del tráfico en la red debido a que los datagramas salientes tienen un tamaño menor que el que se obtendría si se incluyera el certificado en cada uno de ellos.

En cada nodo receptor mcast, el agente espera recibir los datos junto con el certificado digital y si sólo recibe los datos, entonces realiza una búsqueda del certificado correspondiente al usuario que envió los datos en un repositorio de certificados de los nodos que enviaron información anteriormente. Si el certificado digital no existe, el agente descarta el datagrama entrante, en otro caso, realizará la autenticación y verificación de integridad de los datos recibidos. Este repositorio auxiliar de certificados digitales se depura con cierta regularidad, por defecto establecida en 15 minutos, pero susceptible de ser modificada por el administrador de cada nodo receptor mcast.

6.4.3. Validez de los Datos y de Certificados

El modelo seguro permite estimar el *tiempo de validez de los datos* relativos al mismo servicio GRAM que se encuentra registrado en el servicio *Index Mcast*. Cuan-

do el nodo receptor mcast recibe el mismo servicio GRAM en un datagrama entrante utilizando el tiempo de validez puede verificar si esos datos se encuentran obsoletos o no. Además, el tiempo de validez de los datos permite controlar si un servicio GRAM no es recibido en un datagrama multicast por un intervalo de tiempo y eliminar los datos relativos al servicio GRAM si se determina que están obsoletos. Entonces, si los datos recibidos han quedado obsoletos con respecto al tiempo de validez para el mismo servicio GRAM, el agente receptor mcast no lo procesa. Por otra parte, los datos obsoletos, en cada nodo receptor mcast, se eliminan automáticamente cuando el nodo recolector mcast deja de enviar datos al grupo multicast.

La estrategia utilizada para estimar el tiempo de validez de los certificados digitales mantenidos en cada nodo receptor mcast y usados en la validación de los datos entrantes, es similar al esquema anterior. Se estableció que cada nodo recolector mcast envíe al grupo multicast el certificado digital con el que firmará posteriormente los datos referidos a cada servicio GRAM registrado en su MDS4, con una frecuencia de 5 minutos. Cuando el nodo receptor mcast recibe un certificado en un datagrama entrante, si no lo encuentra en el repositorio de certificados, lo agrega al repositorio. Si el certificado ya existe en el repositorio, el certificado es actualizado con el que se acaba de recibir. Si luego de cierto intervalo de tiempo, por defecto 15 minutos, el nodo receptor mcast no recibe un certificado proveniente del mismo nodo recolector mcast, el certificado se considera obsoleto y se elimina del repositorio.

Capítulo 7

Implementación

Se desarrollaron dos implementaciones por cada modelo de organización de servicios Índices, una correspondiente al agente recolector mcast y otra al agente receptor mcast, tomando como modelo una arquitectura cliente/servidor.

Agente Recolector Mcast. Tanto para el modelo seguro como para el modelo no seguro, el agente recolector mcast es una aplicación desarrollada con el lenguaje de programación Java que implementa Socket Multicast. Esta aplicación toma el rol de servidor en cuanto a que, como fuente emisora de tráfico multicast IPv6 *sirve* o envía al grupo multicast, de nodos receptores mcast, cada determinado intervalo de tiempo, información que contiene el servicio Índice del nodo donde se encuentra ejecutándose. En el modelo seguro la aplicación hace uso de llamadas al sistema operativo para ejecutar comandos de la biblioteca OpenSSL que permiten firmar digitalmente los ficheros que posteriormente se transmiten al grupo multicast IPv6.

Agente Receptor Mcast. Tanto para el modelo seguro como para el modelo no seguro, el agente receptor mcast es una aplicación desarrollada con el lenguaje de programación Java que implementa Socket Multicast. Esta aplicación toma el rol de cliente en cuanto a que, como miembro del grupo multicast IPv6, espera recibir información multicast proveniente de los nodos recolectores mcast cada determinado intervalo de tiempo. En el modelo seguro la aplicación hace uso de llamadas al sistema operativo para ejecutar comandos de la biblioteca OpenSSL que permiten verificar la integridad y autenticación de la información recibida en cada datagrama entrante. Posteriormente este agente procesa esos datos para alimentar al servicio *Index Mcast* del nodo donde se ejecuta.

Verificadores de Validez Se desarrollaron dos aplicaciones codificadas en Java, ambas instaladas en cada nodo receptor mcast. El objetivo de estas aplicaciones es mantener actualizada la información referida a los servicios GRAM en el servicio *Index Mcast*, y a los certificados digitales de los nodos recolectores mcast cuyos agentes ya han enviado información al grupo multicast. Estas aplicaciones son configuradas por el administrador para controlar el tiempo de validez de estos datos y ejecutarse con cierta frecuencia de forma tal que la información sea depurada en caso de que los nodos recolectores dejen de transmitir información.

Capítulo 8

Experimentos

8.1. Metodología Experimental

Realizamos una serie de pruebas para verificar que la información referente a los servicios GRAM registrados en varios servicios Índices, según se describió en la Capítulo 6, se transmite firmada digitalmente, sobre multicast IPv6 a un grupo de nodos donde residen los servicios *Index Mcast*. Se midió el consumo de memoria, tiempos de CPU y tiempo entre consultas para una cantidad variable de servicios GRAM, tanto en el nodo recolector mcast como el nodo receptor mcast. Las mediciones se tomaron con respecto al tratamiento de todos los servicios GRAM registrados en el servicio Índice consultado en el nodo recolector mcast.

Este conjunto de pruebas permitió estimar el valor de referencia para el tiempo de validez de la misma información contenida en cada datagrama procesado por el agente receptor mcast para alimentar a un servicio *Index Mcast*.

8.1.1. Plataforma de Pruebas

Los experimentos se ejecutaron sobre dos nodos dedicados con soporte IPv6, integrantes de la red CyTED-Grid y ubicados en el laboratorio del Dpto. de Arquitectura de Computadores y Automática. Uno de ellos se configuró con el agente recolector mcast y en el otro con el agente receptor mcast.

En el nodo receptor mcast se configuró una fuente agregadora de ejecución basada en un proveedor de información que obtiene los datos generados por el agente para alimentar al servicio *Index Mcast*. Los agentes se ejecutaron usando J2RE v1.6. El nodo recolector mcast tenía un procesador Intel P4/Xeon de 2GHz, 1GB de

memoria RAM y Debian 4. El servicio Índice en este nodo fue poblado con entradas de prueba consistentes en elementos XML con información de servicios GRAM. El tamaño de cada entrada fue de ≃3KB. El nodo receptor mcast tenía un procesador Intel(R) Pentium(R) 4 de 3GHz, 2GB de memoria RAM y Debian 4. Los nodos se conectaron mediante una red Ethernet de 100Mbps.

Se ejecutaron las pruebas con el servicio Índice del nodo recolector mcast conteniendo entradas que se incrementaban en 10 unidades por prueba desde 10 hasta 120 entradas las que se procesaron y enviaron a través de multicast IPv6. El tamaño de cada entrada, una vez procesada, osciló entre 2KB y 4KB. Se realizaron 100 ejecuciones por vez del agente recolector mcast para consultar al servicio Índice, procesar y enviar 10, 20, ..., 120 datagramas sin intervalo de espera entre un datagrama y el siguiente (E=0). Luego se repitieron las 100 ejecuciones pero con un intervalo de espera aleatorio entre 0 y 1 segundo (E=[0...1]) entre el envío de un datagrama y el siguiente. De esta forma se intentó simular el tráfico real en cuanto a datagramas entrantes que el nodo receptor mcast debería procesar si existiesen varios nodos recolectores mcast enviando información al grupo.

Realizamos 50 ejecuciones del agente receptor mcast para procesar cada grupo de datagramas entrantes cuando el agente recolector mcast enviaba 10, 20, ..., 120 datagramas para E=0 y E=[0...1].

Consideramos que los servicios Índices en el nivel de site de una VO podría concentrar una cantidad considerable de información correspondiente a servicios GRAM y asumimos el valor de 120 como cantidad límite de servicios GRAM registrados en un conjunto de servicios Índices para la realización de las pruebas.

8.1.2. Configuración del Nodo Receptor Mcast

En el nodo receptor mcast, cada socket tiene asignado un buffer para enviar paquetes y uno para recibir los paquetes. Estos buffers tienen determinado un tamaño por defecto que depende de parámetros del sistema operativo. Para dar soporte a la alta tasa de datagramas entrantes en el nodo receptor mcast, se aumentó el tamaño del buffer del socket para recibir paquetes en este nodo. Se recomienda un valor de 600KB para permitir el proceso de 120 datagramas entrantes. En Linux, el comando para realizar esta configuración es `sysctl`. Los valores recomendados son:

```
sysctl -w net.core.rmem_default=600000
sysctl -w net.core.wmem_default=600000
sysctl -w net.core.rmem_max=600000
sysctl -w net.core.wmem_max=600000
```

8.2. Resultados

En esta Sección se intentará responder a los objetivos planteados anteriormente. Como se mencionó en el Sección 8.1.1, los resultados de rendimiento se obtuvieron sobre dos nodos. Este estudio representa una primera aproximación de una organización de servicios Índices más compleja, comunicada por medio de multicast IPv6 que interconecte nodos geográficamente distantes. Se analizarán los resultados obtenidos y presentaremos algunas conclusiones preliminares que podrían extrapolarse a una infraestructura real.

8.2.1. Consumo de Recursos en los Nodos

Se midió el consumo de tiempo de procesador y el consumo de memoria en el nodo recolector mcast cuando el agente consultaba al servicio Índice, generaba los ficheros, los firmaba y enviaba al grupo multicast IPv6, y en el nodo receptor mcast cuando el agente recibía, validaba y procesaba cada uno de ellos. En los experimentos comparamos el consumo de memoria y tiempo de CPU cuando el agente recolector mcast transmitía con intervalo de espera entre el envío de un datagrama y otro y cuando los enviaba sin intervalo de espera, también cuando aplicaba firmas digitales a los datos o no antes de enviarlos al grupo multicast de servicios *Index Mcast*.

La Figura 8.1 muestra el tiempo de CPU utilizado por los agentes recolector y receptor mcast mientras procesaban los datos según se describió en la Sección 6.3, con y sin soporte para certificados digitales. En el nodo recolector mcast, el tiempo de CPU corresponde al tiempo que consumió este agente para procesar cada consulta al servicio Índice cuando éste tenía 10, 20, ..., 120 servicios GRAM registrados y luego enviarlos dentro de datagramas, considerando para los envíos intervalos de espera E=0 y E=[0...1].

En el nodo receptor mcast, el tiempo de CPU se refiere al tiempo consumido por dicho agente para procesar cada grupo de datagramas entrantes y proporcionar información al servicio *Index Mcast*, cuando el agente recolector mcast envió los datagramas a la dirección de grupo multicast con y sin intervalos de espera.

El tiempo de CPU presentó la misma tendencia en ambos nodos recolectores mcast, cuando los agentes se ejecutaron con y sin soporte para certificados digitales. Es decir, que en el nodo recolector mcast se observó que el tiempo de CPU es *proporcional* a la cantidad de servicios GRAM registrados en el servicio Índice. Esto se debe a que el agente consulta al servicio Índice por todos los servicios GRAM registrados en él, luego analiza dicha respuesta, creando un archivo XML por cada

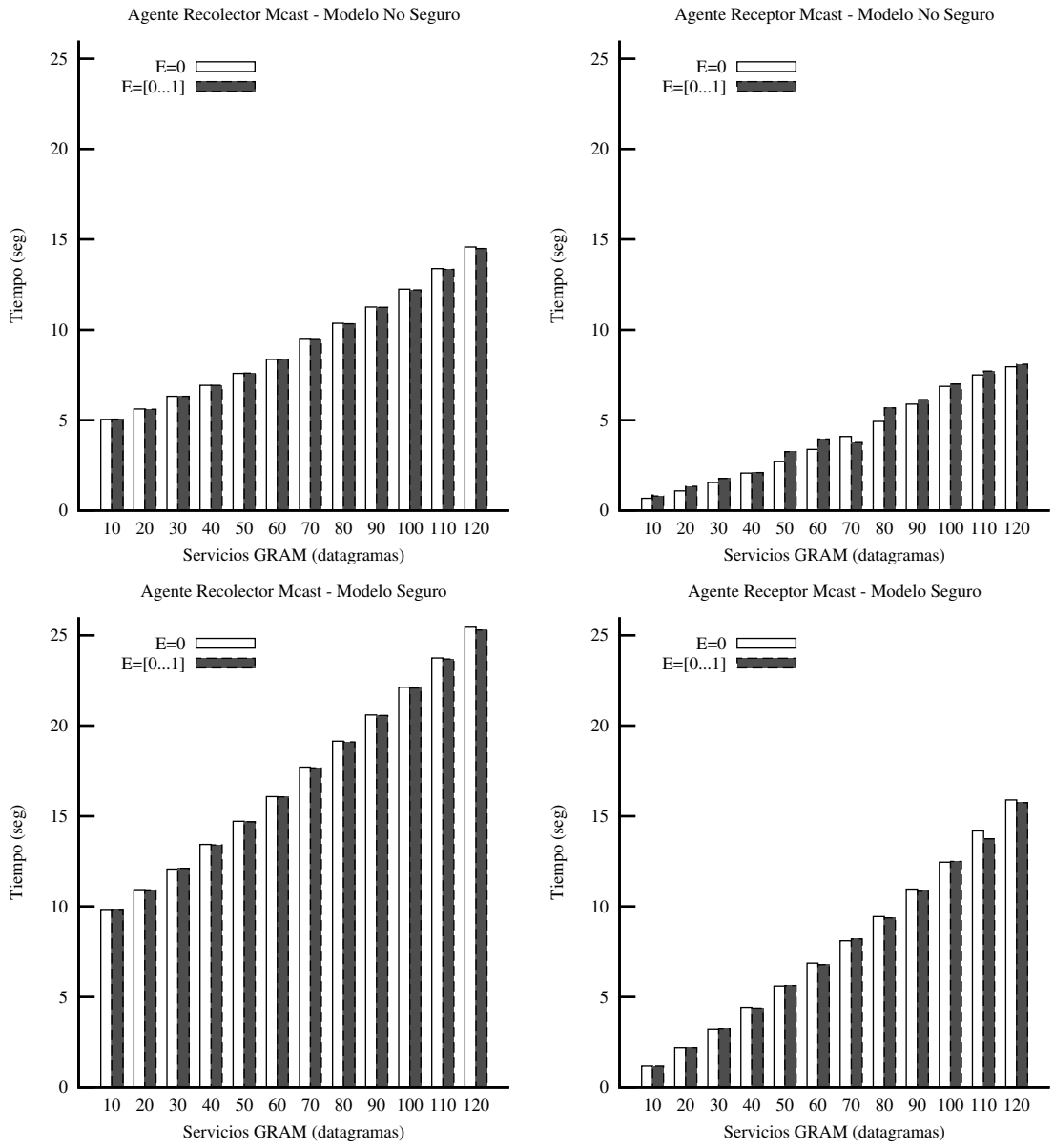

Figura 8.1: Tiempo de CPU en nodos recolector y receptor mcast.

uno de ellos, los que posteriormente se firman (modelo seguro), comprimen junto con el resumen (modelo seguro) y envían a la dirección de grupo multicast. También se observó que el tiempo de CPU es *independiente* del intervalo de espera entre el envío de un datagrama y otro.

En el nodo receptor mcast se observó que el tiempo de CPU necesario para procesar el grupo de datagramas entrantes es *proporcional* a la cantidad de datagramas enviados, pues el agente receptor mcast ejecuta la autenticación y verificación de in-

tegridad de los datos contenidos en cada datagrama entrante, en el caso del modelo seguro y si el paso anterior fue correcto, el agente verifica la validez de la información referente a cada servicio GRAM contenido en el datagrama entrante antes de incluirlo como información válida en el servicio *Index Mcast*.

Se observó que el tiempo de CPU es mayor en el agente recolector mcast con respecto al tiempo de CPU en el agente receptor mcast debido a que en el primero, el agente debe realizar la consulta al servicio Índice, analizar la respuesta, crear un fichero XML por cada bloque correspondiente a un servicio GRAM, firmarlo digitalmente (modelo seguro) empaquetar el fichero XML junto con el resumen firmado (modelo seguro), comprimir y finalmente enviar los datos al grupo multicast.

También pudimos observar en el nodo recolector mcast que el consumo de tiempo de CPU aumentó más rápidamente y en mayor proporción cuando el agente recolector procesaba y enviaba 10 datagramas con y sin firmas digitales que cuando hacía lo propio con 120 datagramas. Es decir, el tiempo de CPU se duplicó al aplicar firmas digitales a 10 datagramas, en cambio, aumentó 83 % cuando tuvo que procesar y firmar digitalmente 120 servicios GRAM registrados.

En el nodo receptor mcast, sin embargo, observamos que a medida que aumentaba la cantidad de servicios GRAM registrados en el servicio Índice, el incremento en el tiempo de CPU entre la solución sin seguridad y la solución segura, fue menor que la misma situación en el nodo recolector. Los valores correspondientes a los tiempos de CPU se incrementaron en 75 % y 80 % para 10 y 120 servicios GRAM recibidos respectivamente.

Por último, con respecto a tiempo de CPU se pudo observar que en la implementación del modelo no seguro, el incremento en tiempo de CPU fue menos significativo entre el envío de 10 y 120 datagramas que en la implementación del modelo seguro, donde el aumento fue más pronunciado.

Los experimentos anteriores nos permitieron obtener valores de referencia para futuras optimizaciones en el consumo de tiempo de CPU en ambos tipos de nodos mcast.

Con respecto al consumo de memoria, los resultados experimentales demostraron que no es significativo en ninguno de los nodos. Se observó que el consumo de memoria RAM fue $\simeq$4MB por encima del consumo que requiere la propia máquina virtual de Java. Este valor se mantuvo durante la ejecución del ambos modelos y fue independiente de la cantidad de servicios GRAM procesados.

8.2.2. Tiempo de Validez de la Información y Certificados

El *tiempo entre consultas* en el nodo recolector mcast es el tiempo transcurrido entre el instante en que el agente recolector mcast consulta al servicio Índice de MDS4, procesa la respuesta, envía los datos al grupo multicast y el instante en que el agente termina de enviar todos esos datagramas.

El *tiempo entre consultas* en el nodo receptor mcast es el tiempo transcurrido entre el instante en que el agente receptor mcast recibe cada datagrama, lo verifica (modelo seguro) y procesa los datos en él contenidos y el instante en que los mismos datos llegan en otro datagrama.

La Figura 8.2 muestra el tiempo entre consultas en el nodo recolector mcast, para ambos modelos, cuando el servicio Índice de MDS4 tenía 10, 20, ..., 120 servicios GRAM registrados para intervalos de espera entre envíos E=0 y E=[0...1].

Se observó que el tiempo entre consultas fue *proporcional* a la cantidad de servicios GRAM registrados en el servicio Índice. Este valor, se aumentó aún más rápido cuando el tiempo de espera entre envíos fue E=[0...1] que cuando E=0.

En el nodo recolector mcast el tiempo entre consultas aumentó más rápidamente y en mayor proporción cuando el agente recolector procesaba y enviaba 10 datagramas con y sin certificados digitales que cuando hacía lo propio con 120 datagramas tanto para E=0 como para E=[0...1]. Es decir, el tiempo entre consultas se incrementó 97 % para E=0 y 45 % para E=[0...1] al aplicar firmas digitales a 10 datagramas. Sin embargo, aumentó 75 % (E=0) y 22 % (E=[0...1]) cuando tuvo que procesar cada datagrama de 120 servicios GRAM registrados.

Los tiempos entre consultas al servicio Índice del MDS4 crecieron cuando el intervalo de espera entre el envío de los datagramas se incrementó. En el modelo no seguro, hemos observado que se duplicó cuando el MDS4 tenía registrados 10 servicios GRAM y aumentó casi 5 veces más cuando tenía 120 servicios GRAM registrados. En el modelo seguro, el tiempo entre consultas sólo aumentó el 50 % en el caso de 10 datagramas a enviar, sin embargo, aumentó 3 veces más para el caso de 120 datagramas a enviar.

De estos resultados se puede concluir que si se aplicase un intervalo de espera entre consultas al servicio Índice y no entre envíos de cada datagrama, se disminuiría la sobrecarga en el tráfico de la red y en el propio servicio Índice del nodo recolector mcast, además de que se evita consultar el estado y la disponibilidad de los servicios GRAM que pueden no haberse modificado en un lapso de 15 segundos o 30 segundos,

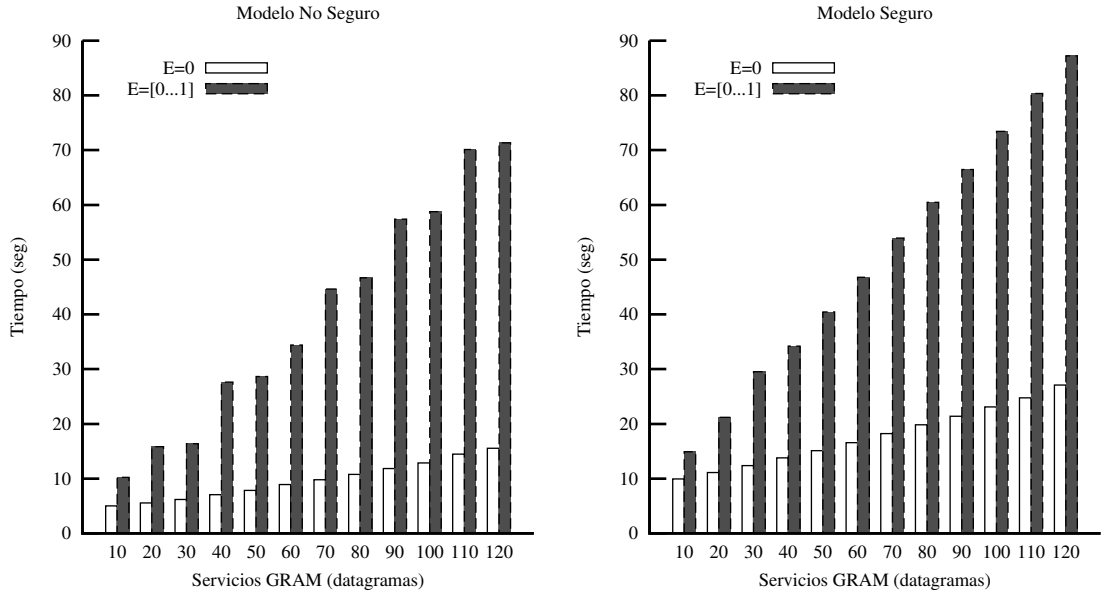

Figura 8.2: Tiempo entre consultas del agente recolector mcast.

es decir el tiempo máximo entre consultas para E=0 en ambos modelos, como se observa en la Figura 8.2.

La Figura 8.3 muestra el tiempo entre consultas, para ambos modelos, que tomó al agente receptor mcast procesar cada grupo de datagramas entrantes y proporcionar información al servicio *Index Mcast* cuando el agente recolector mcast envió los datagramas a la dirección de grupo multicast para intervalos de espera entre envíos E=0 y E=[0...1].

Se observó que el tiempo entre consultas por grupo de datagramas entrantes en el nodo receptor mcast aumentó a medida que el nodo recolector mcast incrementaba la cantidad de datagramas enviados. Este incremento fue mayor en el caso del agente basado en el modelo seguro porque debía verificar la validez de la información referente a cada servicio GRAM contenido en el datagrama entrante antes de incluirlo como información válida en el servicio *Index Mcast*. Asimismo se vió que el tiempo entre consultas es *independiente* del intervalo de espera con que el agente recolector mcast envía los datagramas, pues las operaciones que realiza el agente receptor mcast no dependen de la frecuencia con la que reciba los datagramas, sino de la cantidad de servicios GRAM que tenga registrado el servicio *Index Mcast* al momento de procesar cada datagrama entrante.

Se observó en el nodo receptor mcast que el tiempo entre consultas aumentó más rápidamente y en mayor proporción cuando el agente receptor recibía y procesaba 10

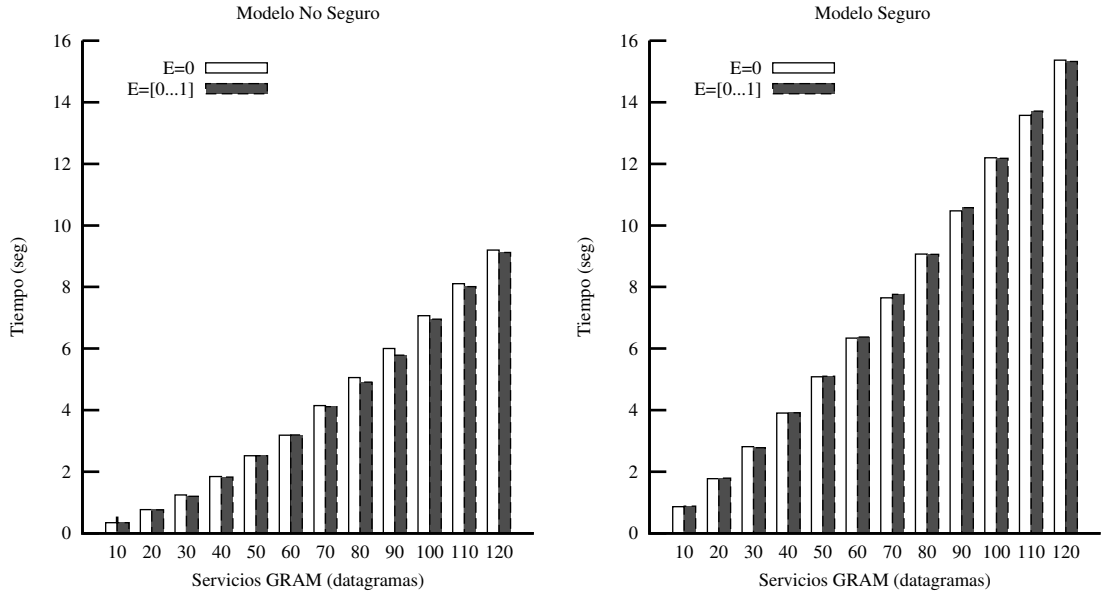

Figura 8.3: Tiempo entre consultas del agente receptor mcast.

datagramas con y sin firmas digitales que cuando hacía lo propio con 120 datagramas para E=0 y E=[0...1]. Es decir, el tiempo entre consultas se incrementó alrededor dos veces y media (147 %) al aplicar firmas digitales a 10 datagramas, en cambio, aumentó un poco más de la mitad (67 %) cuando tuvo que procesar 120 servicios GRAM registrados.

8.2.3. Determinación de Tiempo de Validez de Información

El tiempo entre consultas en ambos nodos mcast nos permitió estimar el tiempo aproximado de validez de los datos contenidos dentro de cada datagrama entrante en el nodo receptor mcast de manera tal que cada servicio *Index Mcast* responda con información actualizada a consultas realizadas por el *Information Manager Driver* de GridWay. Pudimos estimar que un valor de referencia válido de ≃120 segundos representa un tiempo mínimo aceptable para mantener los datos correspondientes a servicios GRAM en cada servicio *Index Mcast*, si tenemos en cuenta el tiempo entre consultas en ambos nodos mcast para procesar 120 servicios GRAM registrados, la latencia de la red, las características de la transmisión multicast en una infraestructura de red real y la configuración adecuada del nodo receptor mcast.

Para calcular dicho valor, se utilizó la Ecuación 8.1.

$$\left(t_{\text{cRecolector}} + latencia + t_{\text{cReceptor}}\right) <= t_{\text{Validez}} <= t_{\text{MaxValidez}} \tag{8.1}$$

Donde:

$t_{\text{cRecolector}}$ es el tiempo entre consultas que consume el agente recolector mcast cuando consulta al servicio Índice conteniendo cierta cantidad de servicios GRAM registrados y que luego envía como datagramas al grupo multicast.

latencia representa el tiempo de latencia de la red, cuyo valor se consideró $\simeq$1000 milisegundos.

$t_{\text{cReceptor}}$ es el tiempo entre consultas que consume el agente receptor mcast desde que recibe cierta información en un datagrama hasta que vuelve a recibir la misma información en otro datagrama posterior.

t_{Validez} corresponde al tiempo de validez de referencia de un mismo conjunto de datos XML correspondiente a un servicio GRAM, es decir, el tiempo que debería permanecer registrado ese servicio GRAM en el servicio *Index Mcast* antes de ser eliminado por considerarse obsoleto.

$t_{\text{MaxValidez}}$ es el tiempo máximo de validez de un mismo conjunto de datos XML referido a un servicio GRAM, corresponde a un valor establecido por el administrador del nodo donde se configure el servicio *Index Mcast*. Es necesario fijar este tiempo máximo de validez de la información para evitar que la información registrada en el servicio *Index Mcast* sea considerada como información válida por un intervalo de tiempo que puede ser excesivo. Por defecto se estableció como límite máximo 10 minutos.

Por último, para calcular el tiempo de validez de la información se realizaron las siguientes suposiciones:

- El valor para $t_{\text{cRecolector}}$ utilizado correspondió al tiempo que consume el nodo recolector mcast para transmitir 120 datagramas con E=[0...1] ($\simeq$90 segundos). Este tiempo es similar al que se obtendría si el agente recolector mcast utilizara intervalos de espera entre consultas al servicio Índice de $\simeq$60 segundos y luego enviase los datagramas con E=0 ($\simeq$30 segundos, según la Figura 8.2). Entonces, para estimar el tiempo de validez de la información asumimos que $t_{\text{cRecolector}}$ tiene un valor $\simeq$90 segundos.
- Dado que en un entorno real de transmisión de datos mediante multicast IPv6, los nodos receptores mcast estarán obligados a procesar una alta tasa de datagramas entrantes, tomamos como base para el tiempo $t_{\text{cReceptor}}$ el tiempo que le llevaría procesar 120 datagramas diferentes antes de volver a recibir un datagrama con la misma información, es decir $\simeq$15 segundos, como se puede ver en la Figura 8.3.

Capítulo 9

Conclusiones y Futuros Trabajos

Un sistema Grid es extremadamente dependiente de la información respecto a la disponibilidad y estado de los recursos heterogéneos, geográficamente distantes y altamente dinámicos que lo integran. Por otra parte, GridWay utiliza los Servicios de Información para planificar el flujo de ejecución óptimo para los trabajos en sistemas Grid. En este contexto hemos implementado un modelo de organización de servicios Índices para MDS4 basado en técnicas multicast IPv6 y firmas digitales para facilitar a GridWay información sobre una mayor cantidad de recursos, que se encuentra accesible desde cualquiera de los MDS4 del grupo multicast.

Los resultados preliminares demuestran que la disposición de servicios Índices utilizando multicast IPv6 permite agruparlos de manera flexible, en cualquier nivel jerárquico posibilitando que la información mantenida y monitorizada por ellos se organice en una estructura plana, redundante, tolerante a fallas, que no sobrecarga en exceso el tráfico en la red y es escalable. De esta manera se garantiza la disponibilidad de información a la que puede acceder GridWay.

A pesar de la inherente limitación del protocolo UDP en cuanto a que no garantiza la llegada de los datagramas al destino, nuestro modelo de organización de servicios Índices se construyó de manera tal que la información contenida en cada datagrama constituye una unidad válida de información para cada MDS4 del grupo multicast.

A nuestro prototipo inicial, se agregaron componentes de seguridad para validar el origen de la información enviada al grupo multicast. Hemos utilizado los componentes de GSI de GT4 para realizar tanto la autenticación de la fuente emisora de los datos como la verificación la integridad de los mismos.

A partir de los resultados obtenidos con los experimentos podemos concluir que es posible mejorar la comunicación entre los servicios de información en entornos Grid utilizando para ello la transmisión multicast IPv6. En particular, el metaplanificador GridWay obtiene beneficios directos ya que la información acerca de servicios GRAM se puede transmitir desde servicios Índices en el nivel de *site* de una VO a un grupo multicast de servicios Índices. De esta manera, GridWay puede acceder a servicios Índices que presentan las siguientes características:

- Poseen una organización plana y redundante, que evita la existencia de cuellos de botella, pues todos los servicios Índices del grupo cuentan con la misma información y pueden ser consultados indistintamente.
- Evitan la presencia de puntos centrales de falla ya que la información se encuentra replicada en cada servicio Índice del grupo multicast.
- Permiten que la información permanezca accesible en cualquier otro servicio Índice del grupo multicast en caso que alguno de ellos falle y quede inaccesible.
- Contienen información actualizada, pues la velocidad de propagación de la información es mayor cuando se utiliza transmisión multicast que en el caso de distribuir la misma información a varios destinos unicast a la vez.

Finalmente podemos concluir que el tiempo de validez de la información, estimado a partir de los tiempos entre consultas en los nodos mcast, es un valor que no resulta excesivo para mantener la información en cada servicio Índice perteneciente al grupo multicast, teniendo en cuenta que los datos se transmiten de manera segura. Por todo lo dicho, creemos que GridWay, de esta manera, accederá a información de mejor calidad y que se encuentra disponible en una organización de servicios Índices para MDS4 basada en multicast IPV6 y que soporta firmas digitales.

9.1. Futuros Trabajos

Se proyecta implementar el modelo de organización de servicios Índices en *sites* participantes de VOs intercontinentales y analizar su comportamiento en un entorno Grid real.

Paralelamente y una vez que la Universidad haya concluido el proceso de implantación de IPv6 en su insfraestructura de red, se tiene pensado realizar los experimentos utilizando el protocolo de encaminamiento multicast MRD6 como plataforma base para las pruebas y conectarse a M6BONE a través de IPv6 nativo.

Bibliografía

[1] A. Adams, J. Nicholas, and W. Siadak. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). Internet Request for Comments RFC 3973. http://www.ietf.org/rfc/rfc3973.txt, 2005.

[2] A. Ballardie. Core Based Trees (CBT) Multicast Routing Architecture. Internet Request for Comments RFC 2201. http://www.ietf.org/rfc/rfc2201.txt, 1997.

[3] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing. Internet Request for Comments RFC 2189. http://www.ietf.org/rfc/rfc2189.txt, 1997.

[4] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 205–217, New York, NY, USA, 2002. ACM.

[5] M. P. Barcellos, M. Nekovee, M. Daw, J. Brooke, and S. Olafsson. Reliable Multicast for the Grid: a comparison of protocol Implementations. http://www.allhands.org.uk/2004/proceedings/papers/208.pdf, 2004.

[6] S. Bhattacharyya. An Overview of Source-Specific Multicast (SSM). Internet Request for Comments RFC 3569. http://www.ietf.org/rfc/rfc3569.txt, 2003.

[7] D. Borman, S. Deering, and R. Hinden. IPv6 Jumbograms. Internet Request for Comments RFC 2675. http://www.ietf.org/rfc/rfc2675.txt, 1999.

[8] A. Conta, S. Deering, and M. Gupta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. Internet Request for Comments RFC 4443. http://www.ietf.org/rfc/rfc4443.txt, 2006.

[9] CyTED-Grid. Tecnología GRID como motor del desarrollo regional. http://www.cytedgrid.org/.

[10] K. Czajkowski, I. Foster, and C. Kesselman. Agreement-Based Resource Management. In *Proceedings of the IEEE*, volume 93(3), pages 631–643, CA, USA, 2005. IEEE.

[11] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. Internet Request for Comments RFC 2460. http://www.ietf.org/rfc/rfc2460.txt, 1998.

[12] Distributed Systems Architecture Group. GridWay Metascheduler: Metascheduling Technologies for the Grid. http://www.gridway.org/.

[13] B. Fenner, M. Handley, H. Holbrook, and Kouvelas. I. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised).Internet Request for Comments RFC 4601. http://www.ietf.org/rfc/rfc4601.txt, 2006.

[14] R. Finlayson. IP Multicast and Firewalls. Internet Request for Comments RFC 2588. http://www.ietf.org/rfc/rfc2588.txt, 1999.

[15] A. Forouzan Behrouz and S. Chung Fegan. *TCP/IP protocol suite*. McGraw-Hill, New York, 2007. ISBN: 0-07-111583-8.

[16] I. Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. *Journal of Computer Science and Technology*, 21(4):513–520, 2006.

[17] I. Foster, K. Czajkowski, D.E. Ferguson, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke. Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF. In *Proceedings of the IEEE*, volume 93(3), pages 604–612. IEEE Computer Society, 2005.

[18] I. Foster, J. Geisler, B. Nickless, W. Smith, and S. Tuecke. Software infrastructure for the I-WAY high-performance distributed computing experiment. In *HPDC '96: Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing*, pages 562–571, Washington, DC, USA, 1996. IEEE Computer Society.

[19] I. Foster and C. Kesselman. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. ISBN: 1-55860-475-8.

[20] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, pages 83–92, New York, NY, USA, 1998. ACM.

[21] M. Gasser and E. McDermott. An Architecture for Practical Delegation in a Distributed System. *sp*, 00:20–30, 1990.

[22] GLUE Working Group. Grid Laboratory Uniform Environment (GLUE). http://forge.ogf.org/sf/projects/glue-wg.

[23] Grid-Café. Grid- Café. http://gridcafe.web.cern.ch/gridcafe/index.html/.

[24] M. Handley, I. Kouvelas, and T. Speakman. Bidirectional Protocol Independent Multicast (BIDIR-PIM). Internet Request for Comments RFC 5015. http://www.ietf.org/rfc/rfc5015.txt, 2007.

[25] E. Huedo, R. Montero S., and I. Llorente M. A framework for adaptive execution in grids. *Software: Practice and Experience*, 34(7):631–651, 2004.

[26] K. Jeacle and J. Crowcroft. A multicast transport driver for Globus XIO. In *WETICE '05: Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, pages 284–289, Washington, DC, USA, 2005. IEEE Computer Society.

[27] LCG - Worldwilde LHC Computing Grid Project. Distributed Production Environment for Physics Data Processing. http://lcg.web.cern.ch/LCG/.

[28] G. Malkin. RIP Version 2. Internet Standard 56. http://rfc.net/std56.txt, 1994.

[29] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press, Boca Raton, 1997. ISBN: 0-8493-8523-7.

[30] R. Moreno-Vozmediano. Resource Discovery in Ad-Hoc Grids. In *International Workshop on Grid Computing Security and Resource Management, ICCS 2006*, volume 3994/2006, pages 1031–1038. Springer Berlin/Heidelberg, 2006.

[31] R. Moreno-Vozmediano. Application layer multicast techniques in grid environments. In *EATIS '07: Proceedings of the 2007 Euro American conference on Telematics and information systems*, pages 1–4, New York, NY, USA, 2007. ACM.

[32] J. Moy. Multicast Extensions to OSPF. Internet Request for Comments RFC 1584. http://www.ietf.org/rfc/rfc1584.txt, 1994.

[33] J. Moy. OSPF Version 2. Internet Standard 54. http://rfc.net/std54.txt, 1998.

[34] Open Grid Forum. GGF -IPv6 - Working Group. http://forge.gridforum.org/sf/projects/ipv6-wg.

[35] Open Grid Forum. Open Grid Services Architecture. http://www.ogf.org/documents/GFD.80.pdf.

[36] OpenSSL Project. OpenSSL Homepage. http://www.openssl.org/.

[37] H. Santos. MRD6, an IPv6 Multicast Router. http://fivebits.net/proj/mrd6/, 2007.

[38] P. Savola, R. Lehtonen, and D. Meyer. Protocol Independent Multicast - Sparse Mode (PIM-SM) Multicast Routing Security Issues and Enhancements. Internet Request for Comments RFC 4609. http://www.ietf.org/rfc/rfc4609.txt, 2006.

[39] J. Schopf, I. Raicu, L. Pearlman, N. Miller, C. Kesselman, I. Foster, and M. DÁrcy. Monitoring and Discovery in a Web Services Framework: Functionality and Performance of Globus Toolkit MDS4. Technical Report MCS Preprint ANL/MCS-P1315-0106, Mathematics and Computer Science Division, Argonne National Laboratory, Jan 2006.

[40] Spirent Communications, Inc. White paper-Multicast Routing-PIM Sparse Mode and Other Protocols. http://www.spirentcom.com/documents/1318.pdf, 2003.

[41] W. Stallings. *Comunicaciones y redes de computadores*. Pearson Prentice Hall, Madrid, 2004. ISBN: 84-205-4110-9.

[42] The Globus Alliance. Gt4.0: Information services. http://www.globus.org/toolkit/docs/4.0/info/.

[43] The Globus alliance. The Globus alliance. http://www.globus.org/.

[44] R. Vida and L. Costa. Multicast Listener Discovery Versión 2 (MLDv2) for IPv6. Internet Request for Comments RFC 3810. http://www.ietf.org/rfc/rfc3810.txt, 2004.

[45] D. Waitzman, C. Partridge, and S. Deering. Distance Vector Multicast Routing Protocol. Internet Request for Comments RFC 1075. http://www.ietf.org/rfc/rfc1075.txt, 1998.

[46] G. Waters, J. Crawford, and S. Guan Lim. Optimising multicast structures for grid computing. *Computer Communications*, 27(14):1389–1400, 2004.

[47] V. Welch. Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective. http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf, 2005.