

CAPÍTULO 6

AGENTES SOFTWARE Y REDES ACTIVAS

6.1. INTRODUCCIÓN

En los últimos años se ha producido una dinámica actividad investigadora en el campo de los agentes aplicados a muy diversas actividades. El concepto de agente puede ser estudiado desde muy distintos puntos de vista por lo que no existe un consenso claro en su definición, aunque sí en la idea intuitiva que todos tenemos de este atractivo campo de investigación. Uno de los ámbitos de acción de los agentes es precisamente el de los sistemas de telecomunicación, porque los agentes pueden dotar a las redes de características de las que carecen en la actualidad. Informalmente, puede decirse que un agente es una entidad con objetivos, que alcanza ejecutando una serie de acciones, mediante un dominio de conocimiento y situado en un entorno concreto. De este modo, podríamos acordar que un sistema multiagente es un grupo de agentes situados en un entorno donde puede desarrollarse algún tipo de comunicación entre ellos, bien directamente, o bien a través del entorno como resultado de alguna de las acciones llevadas a cabo por los propios agentes. No obstante, estas dos definiciones pueden resultar demasiado laxas dependiendo del contexto y, por esto, deseamos aclarar una serie de conceptos importantes, así como presentar las clasificaciones más aceptadas de agentes para concluir presentando de forma resumida el sistema multiagente TAP que proponemos para dar soporte a nuestra arquitectura.

Por otro lado nos encontramos con las redes activas basadas en la posibilidad de equipar a los nodos que las constituyen con características que los convierten en elementos activos. Como en el caso de los agentes, falta también el consenso para decidir lo que es una red activa. No obstante, disponemos de criterios para determinar algunas características que nos servirán para reafirmar nuestros objetivos.

Aunque existen muchos puntos de conexión entre el conceptos de agentes aplicados a las redes de comunicaciones y el de redes activas, existe cierta controversia entre ambos campos de aplicación. No obstante, veremos cómo en nuestro caso, con la implementación del sistema multiagente TAP soportado en los conmutadores ATM activos (AcTMs) que hemos diseñado, conseguiremos dotar a la tecnología ATM de esas características activas que le aportan la interacción, autonomía y adaptación de los agentes software que proponemos. Centraremos por tanto nuestro campo de acción en el concepto más general de los agentes software aplicados a las redes, dando así lugar a lo que llamamos redes programables.

El objetivo de este capítulo consiste en la revisión de estos dos campos de investigación, sin entrar en cuestiones encontradas, ni en elecciones concretas en cuanto a los estándares a usar. Así, nos centraremos en el concepto de agentes software, aplicados a las redes de comunicación en general y a ATM en particular, para conseguir que éstas se conviertan en activas. Fundamentamos de este modo el adjetivo activo del nombre de la arquitectura TAP. Para ello este capítulo presenta en primer lugar los fundamentos del concepto de agente, para pasar después a estudiar los sistemas multiagente como sociedades de agentes. El cuarto apartado establece las características de comunicación entre agentes, para pasar después a describir algunas cuestiones relativas a las arquitecturas. Una vez revisadas las propuestas estándares en la materia, se pasa a comentar los beneficios del uso de agentes en el ámbito de las redes. Seguidamente, se realiza una revisión de diversos agentes diseñados para los sistemas de comunicaciones. Estudiados estos aspectos se comentan los puntos de vista de las redes activas; se analiza y caracteriza la arquitectura del Sistema multiagente TAP que proponemos, y el capítulo termina con un apartado de conclusiones.

6.2. CONCEPTOS FUNDAMENTALES SOBRE AGENTES

El paradigma de los agentes software tiene su origen en la década de los años 70 como una corriente de la Inteligencia Artificial Distribuida (IAD). Sin embargo, desde los primeros trabajos sobre agentes y hasta la actualidad, han aparecido varias disciplinas que se han encargado de definir y delimitar su ámbito de acción y, entre ellas, queremos destacar las siguientes:

- Inteligencia artificial, especialmente en el ámbito distribuido.
- Programación orientada a objetos y programación concurrente.
- Telecomunicaciones.
- Diseño de interfaces hombre-máquina.
- Entornos de producción industrial.

Pero el primer problema al que nos enfrentamos es precisamente a la definición del concepto de agente, ya que existen diversas acepciones para esta palabra y no disponemos de una definición única ni definitivamente aceptada, con la consiguiente controversia. En lo referente a la idea intuitiva de lo que representa el término agente parece que el consenso sea mayor, pero -no obstante- vamos a presentar algunas de las definiciones más representativas de la literatura para movernos en un terreno firme desde el punto de vista conceptual.

El Diccionario de la Lengua Española [1] realiza la siguiente definición de *agente*:

“Agente: Del lat. agens, -entis, p. a. de agere, hacer. 1.adj. Que obra o tiene virtud de obrar. 4.m. Persona o cosa que produce un efecto. 5.Persona que obra con poder de otro...”

El Diccionario Webster [2] define la palabra *agent* con las siguientes acepciones:

“.n. 1. a person authorized by another to act on his behalf. 2. one who or that which acts or has the power to act. 3. A natural force or object producing or used for obtaining specific results. 4. An active cause; an efficient cause. 5. One who works for or manages an agency”.

El Agent Technology Glossary del *Object Management Group* (OMG) [3, URL1] destaca la siguiente definición popular:

“Un agente o agente software es un programa software que hace algo, generalmente en nombre de una persona o de otro agente,

- *posiblemente automatizando alguna tarea (por ejemplo indexando),*
- *posiblemente usando cierta inteligencia (por ejemplo planificando o negociando),*
- *posiblemente activado por algún evento que es monitorizado o ejecutado como tarea de fondo haciendo alguna tarea (por ejemplo seleccionando y filtrando información de interés),*
- *posiblemente moviéndose de un sitio a otro (para recopilar información),*
- *posiblemente comunicando o interactuando con el usuario (a través de charla o diálogo),*
- *posiblemente comunicando o interactuando con otros agentes de una forma coordinada y cooperativa,*
- *posiblemente aprendiendo y cambiando su comportamiento a lo largo del tiempo (en respuesta a cambios en el entorno),*
- *posiblemente operando con iniciativa propia y generando informes periódicos”*.

Por otro lado, Franklin y Graesser indican [4]:

“Un agente autónomo es un sistema dentro de un entorno, que siente el entorno y actúa dentro de él siguiendo su propia agenda para conocer el efecto de lo que sentirá en el futuro”.

Además, Hayes-Roth [4] destacan que:

“Los agentes inteligentes realizan continuamente tres funciones: percepción de las condiciones dinámicas de un entorno; acción que afecta a las condiciones del entorno; y razonamiento para interpretar percepciones, resolver problemas, realizar inferencias y determinar acciones”.

Desde un punto de vista más pragmático el IBM's Intelligent Agent Strategy, Aglets [4] concluye:

“Los agentes inteligentes son entidades software que desempeñan una serie de operaciones en nombre de un usuario o de otro programa, con algún grado de independencia o autonomía, empleando algún conocimiento o representación de los objetivos o deseos del usuario”.

A la pregunta de *¿qué es un agente?* ya respondió Carl Hewitt: *“...para la comunidad que investiga en la computación basada en agentes esta es una cuestión tan embarazosa, como lo es para la comunidad dedicada a la inteligencia artificial el responder a ¿qué es la inteligencia?”*. Sin embargo, OMG [3] parte de un planteamiento más simplista indicando que *“un agente es alguien que actúa”*. Aunque también se precisa que para su uso práctico, los agentes deben poseer una serie de propiedades que van a ser comentadas a lo largo de este capítulo.

Ante la situación que acabamos de describir, el mayor inconveniente que encontramos es que estamos ante un término ampliamente usado por muchos investigadores en campos muy diversos, lo que impide poder dar una definición universalmente aceptada. Wooldridge y Jennings [5] responden a la pregunta anterior sin recurrir a definiciones magistrales y se centran en describir las características que deben tener los agentes para ser considerados como tal. Estas características serán comentadas más adelante una vez centrados ciertos aspectos que consideramos de interés.

Especialmente destacable para nosotros es el planteamiento de [6] donde se indica que *“las redes de comunicaciones futuras se conciben más automatizadas, pudiendo ser gestionadas por entidades software, tanto móviles como estáticas, coleccionando información de estado de la red y que aportan la habilidad innata para invocar cambios efectivos en los controladores de los conmutadores sin la interacción explícita de ningún operador humano.”* Este es el modelo planteado por investigadores y desarrolladores de tecnologías¹ como TINA-C (Telecommunications Information Networking Architecture-Consortium) [URL2] y Active Networks [7, URL3]. Para hacer realidad estos modelos, las plataformas de telecomunicaciones intentan abarcar el, relativamente nuevo, paradigma de los agentes software.

No es nuestro principal objetivo el de describir aquí las propuestas que en los últimos años ha realizado la Inteligencia Artificial en el ámbito de la tecnología de los agentes software, pero sí consideramos oportuno aclarar algunos de los conceptos básicos para centrar y afianzar adecuadamente nuestras investigaciones en el contexto adecuado. Nwana presenta en [8], y en otros trabajos previos, los fundamentos y los campos de aplicación de los agentes software. La referencia [6] realiza una detallada descripción de los aspectos más importantes de los agentes inteligentes desde los planteamientos teóricos, hasta sus arquitecturas y lenguajes de implementación. En [9] se actualizan y revisan las últimas arquitecturas en materia de agentes. Existen otras muchas investigaciones y publicaciones relacionadas con los aspectos de la Inteligencia Artificial y la IAD, donde se discuten muy diversas características de los agentes inteligentes. La revisión de toda esta literatura nos permite determinar que el ámbito de las telecomunicaciones ofrece un atractivo campo de aplicación a las técnicas de los agentes software, tanto en el presente como para el futuro.

Pero antes de seguir adelante cabe distinguir entre varios conceptos que tienden a confundirse por sus numerosos puntos de conexión. Trataremos por tanto de aclarar los conceptos de Agentes Software, Agentes Inteligentes, Agentes Móviles, Arquitecturas de Agentes y Sistemas Multiagentes.

Esta sección aclara los conceptos y nociones que permiten definir lo que es un agente software, así como sus aplicaciones en diversos ámbitos de los sistemas de comunicaciones, centrándonos principalmente en la tecnología ATM. El término agente software ha sido adoptado [10] *“como la frase más general para describir el concepto de una entidad software que automatiza algunas de las tareas consideradas como más mundanas o laboriosas para un agente humano”*. No obstante, en no pocos casos se ha complementado el concepto anterior considerando dos variantes o ampliaciones del mismo, como son los llamados Agentes Móviles y los Agentes Inteligentes Cooperativos o Agentes Inteligentes.

Como veremos, existen importantes diferencias entre estas tres variantes que pueden ser usadas para solventar muy distintos problemas. En cualquier caso, la literatura [11] describe también el concepto general de Agentes Móviles Inteligentes como una importante vía de investigación sobre los sistemas de telecomunicación futuros.

En nuestro caso no deseamos realizar una exposición detallada de los aspectos formales y fundamentales de los agentes software sino, más bien, centrarnos en el caso concreto en que nuestras aplicaciones van a hacer uso de agentes software en la activación de gestión automatizada y el control de sistemas de telecomunicaciones aplicados a la tecnología ATM. Publicaciones como [4-6,8-15] presentan aspectos mucho más detallados de los conceptos de agentes, sus amplias áreas de aplicación, su formalismo, etc.

¹ Tecnología puede ser considerada, en este caso, como un conjunto de métodos para lograr ciertos objetivos relacionados.

Otra de las múltiples definiciones de agentes la podemos encontrar en [10], donde se describe un agente software como un *programa ejecutado independientemente capaz de actuar autónomamente sin supervisión humana directa en tiempo de ejecución, siendo capaz de seleccionar acciones cuando se producen eventos esperados o limitadamente inesperados*. Pueden tener diversas habilidades pero, sobre todo, suelen tener la peculiaridad de poder relacionarse dentro de su propio mundo (entorno operacional o dominio de aplicación). El término agente se ha usado también en el ámbito de los sistemas distribuidos para referirse a las entidades usadas para realizar tareas muy concretas. La referencia [5] es la descripción ya clásica para determinar lo que son los agentes software, destacando sus características de habilidad social, autonomía, reactividad y adaptabilidad.

Por otro lado, [12] presenta una interesante taxonomía de los agentes en el ámbito de los sistemas de telecomunicaciones realizando su división en: agentes locales, agentes de red, agentes basados en IAD y agentes móviles.

En líneas generales, cabe destacar que los investigadores de la Inteligencia Artificial opinan que un agente ideal debería tener tres cualidades destacables: posibilidad de cooperar con otros agentes o programas, capacidad de aprendizaje y también autonomía [14]. Sin embargo, y como ya se ha destacado, el consenso no es sencillo en este aspecto y por eso se consideran para la definición de agentes las siguientes características básicas [15]:

- Situación: que permita localizar a los agentes en un entorno determinado.
- Flexibilidad: de forma que las acciones no tengan que estar prefijadas.
- Autonomía: que les permita actuar sin la obligación de requerir la intervención directa del hombre o de otro programa.

Las definiciones y características ya comentadas permiten realizar una clasificación de los agentes con la intención de determinar el ámbito de actuación y aplicación de éstos. En esta línea se han hecho también múltiples clasificaciones de agentes que dependen de los investigadores que las realizan, pero podemos citar la presentada en [URL1] en función de las propiedades que satisface cada agente, destacando que no todos los agentes deben cumplir todas estas propiedades:

- Autonomía: los agentes son proactivos², siendo capaces de tomar la iniciativa para conseguir sus objetivos por sí solos sin requerir iniciación del usuario, confirmación ni notificación. Es decir, pueden actuar sin intervención externa directa, tienen un cierto grado de control sobre su estado interno y sus acciones están basadas en su propio conocimiento. No sólo reaccionan al entorno, sino que también son orientados a objetivos o propósitos propios.
- Movilidad: capacidad de moverse a donde necesitan, posiblemente siguiendo un itinerario.
- Adaptabilidad: cuando el agente es capaz de adaptarse dinámicamente aprendiendo de su entorno ante la incertidumbre y los cambios. Es decir, la capacidad de responder a otros agentes y/o al entorno en un cierto grado. Incluso, formas avanzadas de adaptación permitirán al agente modificar su comportamiento basándose en su propia experiencia.
- Reacción: actuación sobre el entorno mediante un comportamiento estímulo/respuesta.
- Delegación: pueden actuar en representación de alguien o algo.
- Inteligencia: estado formalizado por el conocimiento (creencia, objetivos, planificación, etc.)
- Sociabilidad: capacidad para colaborar en comunidades para conseguir objetivos comunes.
- Racionalidad: capacidad para elegir una acción basándose en objetivos internos o en el conocimiento.
- Coordinación: con capacidad para realizar actividades compartidas con otros agentes del entorno. La actividad de coordinación se realiza a través de planes o por otros mecanismos de gestión.
- Cooperación o colaboración: capacidad para coordinarse con otros agentes para alcanzar objetivos comunes, es decir, agentes no rivales que alcanzan conjuntamente el éxito o el fracaso.
- Competitividad: es decir, el opuesto a cooperación, de modo que el éxito de un agente lleva al fracaso de otros.

² Suele emplearse el término proactivo en contraposición al concepto reactivo que indica que los agentes se limitan a reaccionar ante situaciones previamente determinadas. De este modo, los agentes proactivos podría decirse que son algo más que activos.

- Interacción: interoperación con humanos, otros agentes, sistemas legales y fuentes de información. En suma, la capacidad para comunicar con el entorno y con otros agentes del entorno.
- Impredecibilidad: de modo que pueden actuar sin que se pueda predecir completamente su comportamiento. Es decir, un comportamiento no determinista.
- Continuidad: como procesos de ejecución continua en el tiempo.
- Transparencia y contabilidad: es decir, serán transparentes cuando se requiera, pero ofrecerán un registro de sus actividades si se solicita.
- Carácter: inclusión de estados emocionales y de personalidad.

Por otro lado, la literatura más centrada en el ámbito de la IA suele presentar los agentes divididos en una clasificación a seis niveles:

- Agentes móviles o estáticos.
- Agentes reactivos o proactivos.
- Agentes cooperativos, autónomos, con aprendizaje o con una mezcla de estas características:
 - ✓ Agentes de interfaz: autónomos y con aprendizaje.
 - ✓ Agentes colaboradores: autónomos y cooperativos.
 - ✓ Agentes ideales: autónomos, cooperativos y con aprendizaje.

En cualquiera de los casos los agentes software son entes autónomos (ver *Figura 6.1*) con sensores para percibir (interactuar) entradas de información o eventos procedentes del entorno o de otros agentes y actuadores³ para efectuar las operaciones de salida sobre el entorno u otros agentes. También tienen habilidades computacionales (razonamiento, búsqueda, etc.) y pueden usar sus modelos racionales y de conocimiento para mapear las entradas en salidas pudiendo maximizar sus utilidades (suele medirse el rendimiento de acuerdo a la racionalidad).

Aunque no existe una clara definición del concepto agente podemos decir que, en el ámbito de las tecnologías de la información, los agentes deben cumplir, como poco, las características de autonomía, interactividad y adaptabilidad, comentadas anteriormente, para ser considerados como tales.

En nuestro caso concreto, con la aplicación general de los agentes a las TI y, concretamente, a las TIC, podría hablarse de las siguientes formas de agentes, ya sean software, hardware o ambos:

- Agentes móviles son aquellos capaces de mover su código a un nodo diferente a donde residen, mientras un agente estático permanece en un solo computador o nodo. En determinadas aplicaciones la movilidad puede ofrecer atractivas ventajas, aunque supone también ciertos inconvenientes relacionados con la seguridad del sistema, con el hecho, de que al mover el código de un agente a otro nodo, se consumen recursos en la comunicación y en el nuevo nodo (CPU, memoria, etc.) en el que se va a alojar el agente.

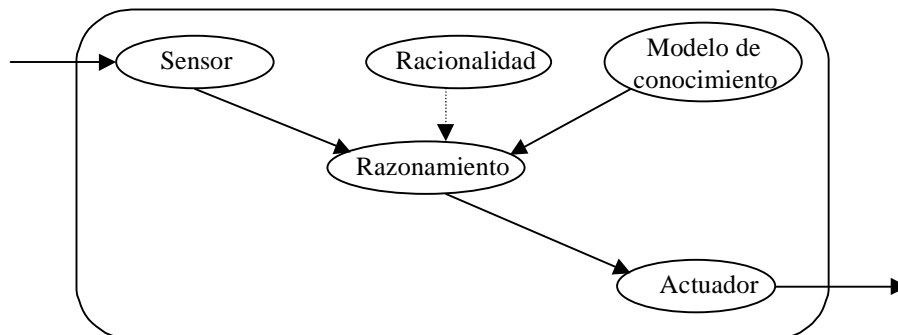


Figura 6.1. Modelo general de agente software

³ Del inglés *effector*.

- Agentes software: son el tipo de agente más específico y, aunque existen también múltiples definiciones, podemos partir de la realizada por OMG [3] en la que se indica que un agente software es una entidad que puede interactuar con su entorno. Es decir, existen agentes que pueden ser implementados usando software, lo que significa que pueden ser autónomos y pueden reaccionar con otras entidades, incluyendo los humanos, máquinas y otros agentes software en varios entornos o plataformas. Debemos destacar que este planteamiento es el que nosotros usamos en nuestra propuesta de forma que nos referiremos a la arquitectura TAP como constituida por agentes software.
- Agentes autónomos: Según la definición de autonomía comentada antes, un agente será autónomo si dispone de independencia de controles externos, es decir, si opera sin intervención ni invocación externa. En realidad, la autonomía se comprende mejor si se especifica en el grado de que se dispone para realizar ciertas acciones. De este modo, suele distinguirse entre autonomía dinámica, y autonomía no determinística, que es impredecible. Un agente es dinámico si es capaz de ejercer algún grado de actividad, pero también puede ser simplemente pasivo o ser completamente proactivo⁴ (pueden elegir abiertamente sus acciones).
- Agentes interactivos son los que cumplen la ya citada característica de interactividad, que también puede ser expresada en determinados grados. El caso más básico de interacción puede ser el método de invocación de los mensajes entre objetos. Y un caso más complejo puede ser la posibilidad que un agente reaccione tras observar ciertos eventos que se producen en su entorno. Esta escala de interactividad puede crecer considerablemente hasta llegar a sociedades de agentes con múltiples interacciones paralelas entre todos ellos con comportamientos competitivos, colaborativos, negociadores, etc.
- Agentes adaptativos son aquellos que, como mínimo, son capaces de reaccionar a un estímulo. Esto es, deben ser capaces de realizar una respuesta predeterminada a un evento particular o señal del entorno. Un sensor de temperatura, o un detector de humos, o un *daemon* de Unix, podrían ser uno de estos ejemplos. Es decir, además de los agentes reactivos podemos tener agentes que pueden razonar. Los agentes que razonan pueden ser también reactivos, sin embargo, se caracterizan por su capacidad para reaccionar realizando inferencias⁵. Las formas más avanzadas de adaptación incluyen la capacidad de aprendizaje y evolución basadas en técnicas de redes neuronales y algoritmos genéticos respectivamente.
- Agentes coordinativos o con la capacidad de coordinar sus acciones con otros agentes para alcanzar un propósito. Cualquiera de las actividades humanas requiere de un trabajo coordinado para conseguir objetivos generales y, por analogía, este planteamiento es también válido en el ámbito de los agentes. Un ejemplo coordinativo puede ser una cadena de montaje de vehículos, o la mayor parte de las actividades de fabricación, pero también se necesita coordinación en la mayor parte de protocolos de comunicaciones para conseguir el objetivo final de cualquier red de comunicación.
- Agentes inteligentes son los que están dotados de inteligencia. Aunque para bastantes investigadores la palabra agente es equivalente a agente inteligente, el mayor inconveniente lo encontramos en la falta de acuerdo en la definición de la palabra inteligencia en el ámbito de la IA. De hecho, el OMG no dispone de una definición normalizada del concepto de agente inteligente, por lo que existen también varios tipos o grados de inteligencia de los agentes. En cualquiera de los casos, se parte de la idea que los agentes requerirán de un conjunto básico de atributos y facultades que se indicaron antes en la capacidad de inteligencia. Un agente inteligente debería ser capaz de examinar sus creencias y deseos, formalizar sus intenciones, planificar las acciones que desea realizar basándose en ciertas suposiciones y actuar para lograr sus planes.
- Agentes envolventes⁶ ofrecen una vía para legar código y aportar la posibilidad de reusar código a un proceso, o transformar un trozo de código estático en una entidad activa. Es decir, este tipo de agentes aporta a otros agentes una capa, interfaz o envoltorio para conectarse a software que no puede ser considerado como agentes. En suma, estos agentes aportan la posibilidad de que los agentes interactúen con las partes de un sistema que no son agentes.

⁴ Aunque las abejas tienen principalmente un comportamiento reactivo, pueden tener también un cierto grado de proactividad cuando eligen comer o volar o picar o reproducirse, etc.

⁵ Sacar una consecuencia de un hecho o de un principio. La deducción es una inferencia.

⁶ Del inglés wrapper para indicar el concepto de interfaz o capa entre diferentes tipos de código.

Se han descrito muchas otras formas de agentes como agentes administradores, facilitadores, colaborativos, etc, pero las que acabamos de comentar son las más ampliamente usadas en la literatura para poder determinar cuándo se está o no ante un agente.

6.3. SISTEMAS MULTIAGENTES (SMA)

La mayor parte de problemas que intenten solventarse con la tecnología de agentes podrían ser solventadas con un solo agente. Sin embargo, lo mismo que el concepto de programación descendente, modular y orientada a objetos intenta dividir los problemas en subproblemas cada vez más sencillos de resolver, en el ámbito de los agentes se huye de la idea del agente omnipotente. Es decir, lo mismo que en la sociedad humana podemos encontrar las actividades claramente divididas para conseguir un elevado grado de especialización y eficiencia, en el caso de los agentes se tiende a este planteamiento en el que múltiples agentes autónomos interactúan, se coordinan (cooperando, compitiendo o ambas cosas) y todo ello de una forma adaptativa. De esta forma se constituyen sociedades de agentes a las que se consideran Sistemas MultiAgentes (SMA).

Por tanto, aunque el uso de un solo agente aparezca como más simple en su implementación, ya que no es necesario prestar atención a las labores de coordinación, negociación, interacción, etc., parece más sensato abordar problemas de cierta escala, como es el caso de nuestra arquitectura de comunicaciones TAP, mediante un sistema multiagente, por, entre otros, los siguiente motivos:

- Aunque podemos tener un agente plenipotenciario, esto puede conducir a claros problemas de escalabilidad y pérdida de eficiencia, por lo que parece lógico dividir las labores y funciones que se desempeñarán entre varios agentes, consiguiendo así modularidad, flexibilidad, robustez, modificabilidad, escalabilidad y extensibilidad.
- El conocimiento concentrado en un solo agente puede conducir a una clara limitación de robustez del sistema, ya que si aparecen problemas en el agente que concentra todo el conocimiento, se propagan a todo el sistema. De este modo, es también intuitiva la ventaja de distribuir el conocimiento entre varios agentes que aporten un cierto grado de tolerancia a fallos.
- La distribución de la carga, de las tareas o de las funciones es otra de las ventajas que pueden obtenerse de los SMA. De este modo, los agentes de un SMA pueden diseñarse como componentes autónomos que actúan en paralelo para resolver problemas de ámbito general.

Como vemos, los agentes pueden ser entes autónomos o bien miembros de un SMA. Resulta relativamente sencillo extrapolar el concepto de agente hacia sistemas, poco o muy complejos, donde intervengan diversos agentes software coordinados entre sí y compartiendo informaciones que les sean de interés. De este modo podemos adquirir una idea clara e intuitiva de lo que sería un sistema o sociedad de agentes múltiples, distribuidos y cooperativos que interactúan entre sí compartiendo información. Las ventajas aportadas por los SMA son muy variadas y algunas de las más interesantes descritas en la literatura han sido compiladas en [10] de donde destacamos y adaptamos a nuestras necesidades concretas las siguientes:

- Pueden usarse para enfrentarse a problemas excesivamente complejos para ser resueltos mediante agentes centralizados por la escasez de recursos o para lograr la optimización de éstos.
- Para ofrecer escalabilidad, ya que los agentes del SMA pueden evolucionar dinámicamente a medida que cambie o evolucione el propio sistema.
- Permiten o facilitan la interconexión con otros mecanismos clásicos, como son los sistemas expertos, los sistemas de ayuda a la toma de decisiones o, principalmente en nuestro caso, con pilas de protocolos de comunicaciones ya existentes.
- Para resolver problemas mediante soluciones provenientes de fuentes de información distribuidas.
- Finalmente, para conseguir las ventajas que todo sistema distribuido pueda aportar y, en suma, donde las características expertas del sistema puedan ser distribuidas en él mismo.

En cualquiera de los casos, los SMA ofrecen importantes ventajas sobre los sistemas de control centralizados, de la misma forma que la computación distribuida mejora las prestaciones de la centralizada.

Parece lógico que los agentes no actúan en solitario, sino que se localizan en entornos o plataformas con varios agentes. Cada uno de los agentes del entorno tendrá sus propios objetivos, tomará sus propias decisiones y podrá tener la capacidad de interactuar y comunicarse con el resto de agentes [42]. Los entornos

o plataformas que soportan varios agentes es lo que en la literatura se conoce con el nombre de SMA o agencias.

El OMG define un sistema multiagente como [32]: “*una plataforma que puede crear, interpretar, ejecutar, transferir y terminar agentes*”. Como características principales de los SMA pueden destacarse las siguientes [15]:

- Cada agente individual del sistema tiene un conocimiento limitado del mismo, es decir, los agentes no disponen de información del sistema completo.
- No existe un control global de todo el sistema por lo que los datos quedan descentralizados.
- La computación es asíncrona en lo que respecta a la interacción entre los diversos elementos del SMA.
- Deben permitir la interoperación con otros sistemas existentes, aunque ésta es de las tareas más complejas en la actualidad.

Sin embargo, hay que destacar también que para poder soportar SMA es necesario disponer de un entorno adecuado para su desarrollo. Este tipo de entornos debe disponer al menos de las siguientes características:

- Contener agentes con las capacidades de autonomía, adaptabilidad, interacción y coordinación.
- Partir de un planteamiento abierto donde se huya de un diseño centralizado.
- Ofrecer una infraestructura donde se pueda especificar claramente la comunicación y los protocolos de interacción entre agentes y con el propio entorno.

Sin embargo [URL4], el diseño del escenario anterior suele plantear una serie de problemas como la elección del tipo de comunicación entre los agentes, del tipo de plataforma o del método de desarrollo, de los criterios para la seguridad del sistema, etc.

En todos estos casos podemos encontrar cómo las redes de comunicaciones son un claro ámbito de aplicación del paradigma de los agentes software distribuidos con sus múltiples variantes. Con la aplicación de las técnicas de agentes, las redes pueden ser más fácilmente administradas, sus recursos pueden ser optimizados ofreciendo QoS, los operadores de las redes pueden ver simplificado su trabajo, los usuarios finales pueden tener nuevos servicios y satisfacer sus necesidades de QoS y los proveedores de servicios pueden incrementar la productividad de sus redes. Éstas son algunas de las ventajas más generales, pero en el resto de este capítulo particularizamos en algunas de ellas.

6.4. COLABORACIÓN, COORDINACIÓN Y COMUNICACIÓN ENTRE AGENTES

Los problemas que intentan solventar los SMA se consiguen resolver en función del grado de cooperación o colaboración que los agentes individuales son capaces de compartir mediante una estrategia de interacción adecuada. Así, es muy importante poner un especial énfasis en el diseño de esta vital faceta de los agentes que tradicionalmente ha presentado tres estrategias de interacción básicas y que vamos a relatar resumidamente a continuación:

- Agentes cooperativos: Trabajan conjuntamente con la intención de resolver juntos un problema y pueden verse como un modelo adecuado para la gestión de redes. En general, esta estrategia es útil para el control de sistemas críticos donde debe conocerse el estado de equilibrio del sistema que, en nuestro caso particular, puede ser el control de errores de una red y sus congestiones. En conclusión, este tipo de agentes está pensado para sacrificar la utilidad individual a cambio del bien general de todo el sistema.
- Agentes auto-interesados son aquellos que intentan maximizar su propio beneficio sin preocuparse del interés general del sistema. Este planteamiento es útil en el caso de las redes cuando aparecen múltiples proveedores de red interesados en la bondad de sus propios servicios. Para nuestro ámbito de actuación puede tener interés aplicar este tipo de comportamiento en la negociación y mantenimiento de los parámetros de la QoS que los usuarios de la red pueden requerir. En general, este tipo de comportamientos son apropiados cuando los agentes van a competir en entornos abiertos, como pueden ser diferentes tecnologías de redes y necesidades de usuarios.

- Agentes hostiles: Este tipo de agentes no tiene una aplicación directa en el caso de los sistemas de telecomunicaciones y se basan en la idea de que tienen una utilidad que se ve incrementada con su propia ganancia y con la pérdida de otros agentes competidores.

Entre estas propuestas se han realizado múltiples investigaciones en el ámbito de la planificación distribuida, la planificación general, la asignación de recursos y el control de problemas mediante agentes cooperativos. Aunque el *Capítulo 7* profundiza en estos aspectos, queremos adelantar que algunas de las ventajas para la distribución son las siguientes:

- La reducción de los costes de comunicación asociados a los sistemas centralizados. Es importante destacar que, para que esta ventaja pueda ser conseguida, debe requerirse el mínimo consumo de ancho de banda para la comunicación inter-agentes.
- Incremento de la velocidad y optimización de cómputo a través de la paralelización.
- Incremento de la reactividad sin necesidad de consulta a entidades centralizadas.
- Robustez, disminuyendo las dependencias de un nodo concreto. En algunos casos esta robustez puede ser también conseguida mediante la replicación de algunos agentes, en lugar de la distribución.

Los agentes que forman un SMA deben colaborar entre ellos para desempeñar sus tareas respectivas. Esta colaboración suele realizarse mediante un lenguaje de comunicación comprensible por todos los agentes del sistema y también por otros programas si es necesario [URL4]. En contraste con un objeto convencional –que ejecuta métodos de otro objeto siempre que tenga permiso–, un agente puede rechazar una petición de otro agente, por lo que deben ser capaces de comunicarse entre sí, para decidir qué acción realizar o qué datos obtener [16]. El mayor inconveniente para conseguir la comunicación está en el hecho que existen múltiples lenguajes propios de cada fabricante, lo que no facilita la compatibilidad en sistemas heterogéneos.

A principios de los años 80, el DARPA estadounidense desarrolló el Knowledge Query Management Language (KQML) [URL5] como una norma para la comunicación entre agentes. Dicho lenguaje incluye múltiples primitivas –llamadas ejecutivas– que definen las operaciones que un agente realiza en su comunicación con cualquier otro. Además, un entorno KQML puede enriquecerse con agentes especiales –llamados facilitadores– que proveen funciones adicionales.

Algunos entornos –como el Java Agent Template (JATLite) de la Universidad de Stanford (EE.UU.) [URL6]– han desarrollado compatibilidad con KQML. Sin embargo, han aparecido varios dialectos KQML con una sintaxis similar, pero no completamente compatibles.

La Fundación para los Agentes Físicos Inteligentes (FIPA) [URL7] propuso como norma a principios de los años 90 el lenguaje Arcol (desarrollado por France Télécom [URL8]). Arcol conserva –al igual que KQML– una sintaxis similar al lenguaje LISP, pero incluye una semántica formal, lo que proporciona una base rigurosa para la interoperación entre agentes y evita la proliferación de dialectos [16]. Sin embargo, Arcol carece aún de algunos aspectos deseables en un lenguaje pensado para la comunicación de agentes, como permitir una mayor autonomía, una mayor heterogeneidad y el uso de dialectos abiertos.

Por otro lado, CORBA pretende definir una norma genérica para la comunicación y la interacción de sistemas multiagentes creados por distintos fabricantes.

Los agentes de un SMA deben coordinar sus actividades para alcanzar sus objetivos y ser capaces de negociar con otros agentes cuando aparecen problemas como la escasez de recursos. La coordinación es, por tanto, necesaria para determinar la estructura organizativa entre un grupo de agentes. Posiblemente, la técnica de coordinación más exitosa para la asignación de recursos y tareas entre una sociedad de agentes es Contract-Net Protocol (CNP) adoptado en las especificaciones de FIPA [URL4]. En CNP se aplica una estructura de mercado donde los agentes toman dos papeles principales, el de director y el de contratista. El principio base para la coordinación es que, cuando un agente no puede resolver la acción que tiene asignada usando su conocimiento local, lo que hará será descomponer el problema en subproblemas y buscará otros agentes con recursos suficientes que sean capaces de solventar los subproblemas obtenidos. La asignación de los subproblemas se solventa mediante un mecanismo de contratación por el cual el agente director crea un contrato que luego difunde (y éste puede ser un inconveniente en entornos de red) a otros agentes del sistema que pueden aceptarlo o no. CNP aporta las ventajas de permitir la asignación dinámica de tareas permitiendo la participación de otros agentes.

La comunicación entre los agentes y la forma de realizarla es otra de las características importantes de los SMA. Los agentes pueden pasar información a otros agentes para compartir entre sí los cambios producidos en el sistema o bien para informarse de los cambios previstos en el estado del sistema. Esto generalmente se realiza en los SMA a niveles elevados de abstracción, en lugar de usar los tradicionales métodos remotos de

paso de mensajes. Normalmente, la comunicación entre agentes implica el paso de información a nivel de conocimiento, y la comunicación puede realizarse a través de mecanismos dinámicos como es la memoria compartida, los IPC o a través de mecanismos estáticos como es el uso de ficheros con bloqueos. De este modo, la literatura presenta dos categorías de comunicación entre agentes; por un lado, mediante el uso de Agent Communication Languages (ACL) propietarios y por otro, mediante el uso de ACL normalizados.

La comunicación permite a los agentes de un SMA el intercambio de información con otros agentes para anunciar su situación actual, recursos disponibles, entorno local, etc. como base para cooperar con otros agentes. Mediante la comunicación de este tipo de informaciones los agentes pueden trabajar de una forma más flexible.

6.5. ARQUITECTURAS DE SMA

Tradicionalmente se ha presentado la arquitectura de agentes dividida en dos niveles generales. Por un lado, el primer nivel define la arquitectura interna del agente, es decir, las relaciones entre los componentes internos de cada agente. En segundo lugar lo que se define como la arquitectura del SMA que presenta las relaciones e interacciones entre cada uno de los agentes individualmente.

La referencia [5] presenta una clasificación de los agentes en función de la arquitectura interna del agente. Así, se pueden encontrar tres tipos de arquitecturas: reactiva, deliberativa e híbrida. Los autores de esta clásica referencia realizan una descripción detallada de estas tres arquitecturas así como de las investigaciones relacionadas con ellas. Por otro lado [9] analiza y discute los ámbitos en que se han aplicado estas arquitecturas internas, así como los campos en que podrán ser aplicadas en el futuro.

Existen otros planteamientos en lo referente al importante aspecto de las arquitecturas de agentes y el trabajo [13] presenta otras arquitecturas internas de agentes basándose en las técnicas de IA que pueden usarse como base de la tecnología de agentes. La *Figura 6.2* presenta una visión básica de agente donde pueden observarse las entradas, salidas y acciones realizadas por los agentes software para interactuar con su entorno externo.

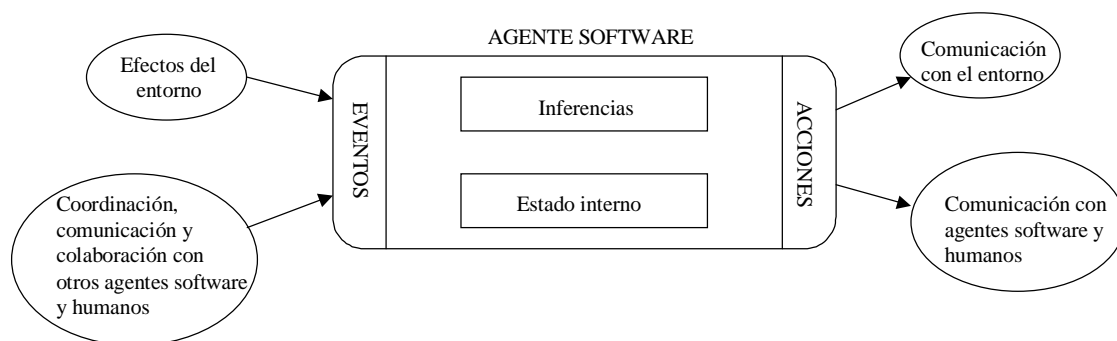


Figura 6.2. Interacción de agente software con su entorno

En [10] se destaca que la adopción de una arquitectura interna de agente específica como oposición a otra arquitectura es inútil si la arquitectura multiagente no está bien definida. Así, el nivel superior de la arquitectura multiagente puede entenderse como un nivel que los agentes necesitan para poder coordinar sus mecanismos de colaboración en el SMA completo. Por tanto, en este nivel se pone especial atención en el grado de cooperación o colaboración que se facilita entre los agentes.

Existen en la literatura muy diversos modelos de SMA que acaban dando lugar a múltiples topologías que representan la situación de los agentes en la red, su interconexión y la forma en que resuelven sus tareas. En cualquiera de los casos suponemos el modelo de agente presentado en la *Figura 6.1*, de forma que estos agentes unidos en grupo constituirán el SMA, siguiendo dos modelos generales: la topología completamente mallada en que todos los agentes están comunicados entre sí, y la topología jerárquica en la que cada agente está solo conectado con su padre (predecesor) y su hijo (sucesor) en la jerarquía. La *Figura 6.3* muestra estas dos topologías en la que puede entenderse que cada agente únicamente se puede comunicar o interactuar con aquellos agentes con los que está conectado. Suponemos que cada uno de los nodos de la red soporta un agente software.

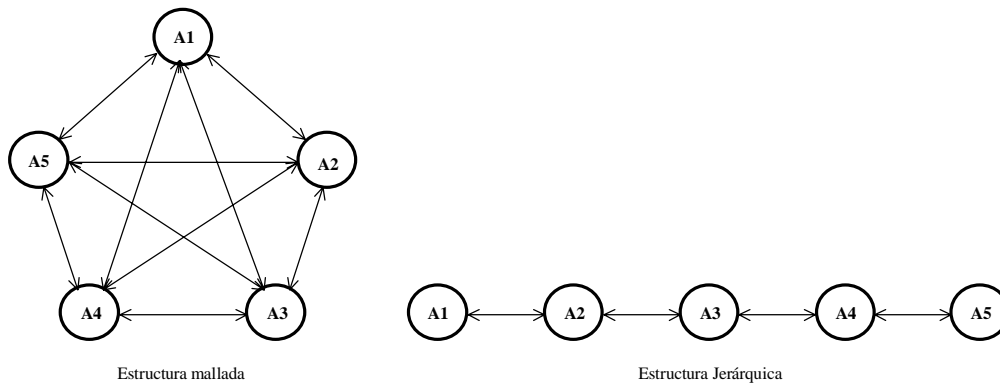


Figura 6.3. Topologías generales de SMA

La elección de un modelo mallado o jerárquico tiene importantes efectos sobre la escalabilidad, el rendimiento y la estabilidad del sistema. Estas tres propiedades no funcionales de los SMA son estudiadas en el *Capítulo 7*, sin embargo, podemos ver ya que es clara la implicación respecto a los aspectos de escalabilidad y conectividad entre los dos modelos que pueden observarse en la *Figura 6.3*. Si nos fijamos en el caso de una red completamente mallada en la que todos los agentes están conectados entre sí, puede calcularse el número de enlaces de una red de N nodos con la expresión

$$\frac{N(N-1)}{2}$$

Como en el caso de las redes de ordenadores, en el modelo mallado de agentes, el número de los nodos del sistema crece linealmente, sin embargo, el número de conexiones entre los nodos crece de forma cuadrática. Por ejemplo, si estamos ante un escenario que está formado por nodos con 6 agentes software cada uno, que deben estar todos completamente mallados, el número de agentes de la red viene dado por la expresión,

$$\frac{6(N(N-1))}{2} = 3N(N-1)$$

Así, la estructura jerárquica reduce la complejidad de computación limitando el número de sus agentes hijo y puede enfrentarse con el incremento de comunicación causado por la incertidumbre. La mayor ventaja es que el entorno puede ser escalable y flexible.

Es importante la consideración de los problemas de la escalabilidad en los sistemas multiagente, aunque éste es un aspecto poco estudiado y, menos aún en el caso de su aplicación a los sistemas de telecomunicaciones. Generalmente las investigaciones se han centrado más en las propiedades funcionales de los SMA donde se estudian las características de los agentes y la funcionalidades que son capaces de aportar. Las referencias [17,18] son dos interesantes estudios que se centran en la investigación de otros aspectos diferentes a los puramente funcionales, principalmente en las propiedades relacionadas con el rendimiento del SMA.

Podemos ver claramente cómo los modelos de SMA se ajustan perfectamente a su aplicación en los sistemas de telecomunicación, por lo que podemos pensar en extender estas dos topologías básicas a cualquiera de las topologías de redes que conocemos para dar lugar a nuevas estructuras de SMA. Por ejemplo, podemos pensar en estructuras de SMA en bus, en anillo, en estrella, en doble anillo y doble bus, en topologías arborescentes y en forma de grafos, en situaciones punto-punto, punto-multipunto y multipunto-multipunto donde cada nodo puede tener varios predecesores y múltiples sucesores, etc. Queremos destacar, por tanto, que podemos encontrarnos con múltiples variantes de los dos casos más comunes y que, sobre todo, hay que destacar la característica de posibilidad de comunicación en función de la conectividad que exista entre los agentes. Por extensión, por delegación o por cualquier otra vía, cualquier agente de un SMA podrá estar conectado con otro agente, lo mismo que cualquier nodo de una red puede comunicar con otro nodo de esa misma red aunque no exista una conexión directa entre ambos nodos. Como veremos, en nuestra propuesta de SMA-TAP estaremos ante una topología jerárquica arborescente para poder soportar las necesidades p-p, m-mp y mp-mp de las redes de tecnología ATM.

6.6. NORMALIZACIÓN DE LA TECNOLOGÍA DE AGENTES

En los últimos años, varios organismos (OMG, FIPA, CLIMATE, KQML, etc.) y proyectos de investigación auspiciados por los gobiernos (ESPRIT, DARPA, ACTS, CoABS) han intentado centrar los diversos aspectos de la interoperabilidad de los agentes, basándose en las siguientes características [19]:

- Debería ser posible acceder a un conjunto de servicios de infraestructuras compartidas, entrega de mensajes fiable, nombrado de agentes, interoperación estructural, etc.
- Debería ser posible compartir una ontología común, como teoría veraz y realista; y métodos de relación entre objetos y variables.
- Debería haber una convención en la sintaxis y semántica de un lenguaje de comunicación de agentes (ACL) común en el que sea posible expresar todo ello.

En cualquier caso, la mayor parte de las iniciativas de interoperabilidad coinciden en que estas tres normas parecen insuficientes, ya que existen múltiples plataformas de agentes que las cumplen pero que no son capaces de interoperar. Un primer problema aparece porque la interoperabilidad entre lenguajes no es suficientemente amplia. Específicamente, para el caso de ACLs como KQML [20] y el FIPA ACL [21], la interoperabilidad de los lenguajes requiere -más que simpleza- la convención de agentes en el formato y significado de las primitivas de los mensajes ACL. En cierto modo, se nota una carencia de especificaciones técnicas que faciliten la “conversación” entre diferentes técnicas de agentes.

En la última década se ha realizado una importante actividad investigadora en el área de la gestión de redes de telecomunicaciones. Muchas de estas investigaciones han culminado en TMN (Telecommunication Management Network) [22] con sistemas basados en los estándares implantados en la actualidad. Iniciativas más recientes como TINA (Telecommunications Information Networking Architecture) [23] han propuesto la gestión integrada con control de servicio para los teleservicios multimedia integrando y desarrollando los conceptos TMN e IN (Intelligent Network). Mientras TMN usa actualmente el modelo manager-agent y los protocolos OSI Systems Management para sus interfaces interoperables, TINA [24] aboga por el uso de tecnologías de procesamiento distribuido de propósito más general como es el OMB CORBA (Common Object Request Broker Architecture) [25].

En la actualidad, los entornos de agentes móviles están dirigidos por dos organismos preocupados por su estandarización. La FIPA (Foundation of Intelligent Physical Agents) [21] cuida de la interacción semántica de alto nivel entre los agentes software que desarrollan alguna forma de inteligencia y adaptabilidad, teniendo sus raíces en la IAD. OMG cuida más de la movilidad de acuerdo a un entorno interoperable estándar a través de su MASIF (Mobile Agent System Interoperability Facility) [26].

En los aspectos conceptuales y funcionales existe un elevado grado de coincidencia, sin embargo, en lo relativo a la normalización de plataformas de desarrollo, lenguajes de comunicación entre agentes, mecanismos de transporte de mensajes, ontologías, protocolos de integración, etc., nos encontramos ante una verdadera torre de Babel que intenta converger hacia una necesaria normalización de la tecnología de agentes. Los organismos más importantes en esta tarea normalizadora son los siguientes:

6.6.1. OMG

OMG (Object Management Group) [3] está constituido por varios Grupos de Trabajo que dirigen sus esfuerzos hacia la normalización de la tecnología de agentes. OMG publica periódicamente RFI (Request For Information) y RFP (Request For Proposal) solicitando opiniones y aportaciones en todo lo relativo a la tecnología de agentes.

6.6.2. FIPA

FIPA (Foundation for Intelligent Physical Agents) [URL7] está trabajando desde 1996 en el área de la estandarización de agentes y tiene múltiples trabajos y documentos desarrollados y en curso con muchas coincidencias con OMG. Sus mayores esfuerzos en la actualidad se centran en la creación de una arquitectura que pueda soportar múltiples mecanismos para la implementación de sistemas de agentes sobre sistemas de computación distribuida como son Java y CORBA.

6.6.3. CLIMATE

CLIMATE (Cluster for Intelligent Mobile Agentes for Telecommunication Environments) [URL9] representa un grupo de proyectos relacionados con la tecnología de agentes. CLIMATE se constituyó en

1998 en el ámbito de la Unión Europea y en el contexto del programa de investigación y desarrollo ACTS (Advanced Communications Technologies and Services) y actualmente lleva a cabo un completo grupo de proyectos que investigan el uso de los agentes en varias áreas de aplicación como es el control de servicio en redes móviles y fijas, la gestión de telecomunicaciones, el comercio electrónico, las aplicaciones multimedia, etc. En este caso, se incorporan las especificaciones estándares de FIPA y OMG.

6.7. VENTAJAS DEL USO DE AGENTES EN COMUNICACIONES

A pesar de la gran actividad desarrollada en este campo en los dos o tres últimos años, la tecnología de agentes está aun hoy en un estado casi embrionario. Su ámbito de aplicación es tan amplio como el de la programación en general. Sin embargo, hemos de destacar que la aplicación de los agentes al campo de las comunicaciones ha sido y es de los más activos. De este modo se han propuesto agentes para asistir o ayudar en labores complejas de administración de redes⁷, para el balanceo de la carga, para anticiparse a los fallos de los sistemas de telecomunicación, para analizar problemas o para sintetizar todo tipo de informaciones relativas a diversos aspectos de las redes. Destacable es el hecho de la rápida evolución de las comunicaciones y computación móvil, donde los agentes -especialmente en su característica de movilidad- aportan importantes propiedades.

Los sistemas multiagentes suelen ser protocolos del nivel de aplicación situados sobre una capa de distribución (ej. CORBA) que ofrecen varias características destacables como son:

- Un nivel de comunicación en la capa de aplicación que es común a todos los agentes.
- Protocolos de interacción que describen cómo los agentes pueden coordinar sus actividades separadas. Los agentes pueden “negociar” para decidir cómo deben actuar.
- Escalabilidad: permitiendo la expansión de una red añadiendo más autoridades en lugar de incrementar la capacidad de los existentes.
- Efectividad: distribuyendo y reduciendo la gestión del número de saltos desde los extremos hasta los nodos activos (que soportan agentes software), lo que minimiza las latencias en la comunicación (QoS ATM).

La coordinación entre nodos activos de la arquitectura propuesta aporta beneficios específicos al control de tráfico en las redes ATM y, entre ellos, podemos destacar:

- El uso más eficiente de recursos (los agentes los optimizan).
- La naturaleza distribuida y adaptativa de los agentes aporta redes más flexibles y escalables.
- Alivian el trabajo de los operadores humanos.
- Los agentes toman decisiones de forma estándar, lo que reduce las dependencias de la integración de fabricantes.
- Pueden tomarse decisiones de conexión en QoS y en otros factores como la caracterización del tráfico en coste.
- Redes más fiables por el control distribuido aportado por los agentes (la arquitectura TAP que proponemos aprovecha especialmente esta característica).

En nuestro caso, con la arquitectura TAP proponemos un nuevo paradigma para la inclusión de agentes software programables en los conmutadores activos ATM. Los agentes que proponemos se ejecutan autónomamente y cooperan coordinadamente todos entre sí para obtener un objetivo general que es la consecución de una red ATM activa programable. Concretamente, nos centramos en un problema puntual donde los agentes se encargan de solventar una situación que las propuestas estándares no resuelven como son las pérdidas de células en situaciones de conmutadores congestionados.

⁷ En el ámbito de las redes TCP/IP el concepto de agente SNMP se ha usado durante años para la realización de labores de gestión de red, pero este tipo de agente sólo coincide en el nombre con los agentes que estamos tratando.

6.8. GESTIÓN CENTRALIZADA Y DISTRIBUIDA DE REDES MEDIANTE SMA

La IAD y la aplicación de conceptos y tecnologías de agentes inteligentes puede ser también usada para unificar las ventajas de la gestión de redes centralizada y distribuida. La IAD promueve el uso de comportamientos más que el control funcional. De este modo, los agentes pueden ser usados para la construcción, mantenimiento y comprensión de sistemas grandes y complejos. En esta línea el trabajo [27] propone un sistema jerárquico de controladores independientes (agentes) con solución de problemas locales y características de construcción de decisiones. Cada agente actúa como sistema de gestión centralizado, pero colectivamente se comunican entre sí para mantener objetivos generales del sistema y para resolver conflictos de interés entre todos ellos. Así pueden obtenerse arquitecturas híbridas aplicables a la gestión distribuida y al control de tareas como la gestión de redes heterogéneas, la gestión de QoS y la administración de administradores de redes.

La asignación de recursos es un problema difícil de resolver en la gestión de redes, principalmente porque el estado operacional de éstas es demasiado dinámico, sobre todo en las redes de banda ancha como ATM. Muchas investigaciones se han dedicado a las estrategias óptimas para el control de admisión de conexión, rutas topológicas, configuraciones de capacidad óptima. Sin embargo, existen muy pocas investigaciones relacionadas con el problema de la reconfiguración topológica que requiera la coordinación de diversos aspectos de la red. Una de las investigaciones en el ámbito de ATM es [28], que usa un agente *peer-to-peer*. Contrastando con lo anterior [29] y [30] proponen una descomposición jerárquica del sistema multiagente y [6] propone un SMA estratificado en múltiples capas.

Otro interesante trabajo es [31], que presenta un agente cooperativo para enfrentarse al problema del control de congestión en las redes ATM. Usa agentes localizados en los buffers de los conmutadores para ofrecer información de realimentación al agente del nodo fuente para que las conexiones que sean añadidas después a la red no se incorporen a la zona congestionada.

Por otro lado, FIPA [URL7] ha definido los estándares para la aplicación de agentes en la gestión de redes. El escenario particular de la especificación usa agentes cooperativos especializados en la negociación para generar una red privada virtual (VPN). Este escenario presenta tres niveles: un agente para la representación de cada usuario humano, conocido como Personal Communications Agent (PCA). El segundo tipo de agente es el Service Provider Agent (SPA) que representa el interés y objetivos del proveedor de servicios y, por último, el Network Provider Agent (NPA). Los PCAs negocian con los SPAs existentes para determinar el mayor nivel disponible en materia de QoS y coste.

Aunque la asignación de recursos es uno de los factores más importantes en la aplicación de la tecnología de agentes en las redes de comunicaciones existen otras propuestas como las siguientes:

- Comunicaciones de un agente con el resto de agentes, con la intención de minimizar esa intercomunicación y hacerlo lo antes posible.
- Estudio de la situación donde deben situarse los agentes, por ejemplo, un agente por nodo, un agente por enlace físico, uno por cada conexión, etc.
- Análisis sobre si los agentes deben ser estáticos o móviles.
- Visión que los agentes pueden tener del resto de la red. Es decir, determinar si los agentes pueden tener o no conocimiento de los eventos locales a su entorno.
- Estudio del grado de dependencia que debe existir entre los agentes de un SMA. Esto permite determinar la robustez del sistema y saber cómo comportarse en el caso de que un agente concreto deje de actuar.

6.9. REVISIÓN DE AGENTES SOFTWARE APLICADOS A LOS SISTEMAS DE TELECOMUNICACIÓN

El trabajo [10] presenta resumidamente los proyectos más interesantes en materia de agentes software en el ámbito de los sistemas de comunicaciones. Cabe destacar que tradicionalmente se han clasificado también los agentes en cuatro categorías principales que son: los basados en SMA; los que emplean la tecnología de agentes móviles; los entornos basados en hormigas y, por último, las propuestas basadas en técnicas económicas.

Se puede decir que las iniciativas en la aplicación de agentes a los diferentes aspectos de las telecomunicaciones, aunque no abundantes, si empiezan a ser significativas. En este apartado comentamos algunas de estas propuestas sin ánimo de ser extensivos ni tampoco profundos en la descripción.

6.9.1. GESTIÓN DE SISTEMAS DE COMUNICACIÓN MEDIANTE AGENTES MÓVILES

El OMG realiza en MASIF [URL7] las siguientes definiciones:

- Agente estático: aquél que sólo puede ejecutarse en la máquina donde fue iniciado. Si éste necesita interactuar con otros agentes o programas o requiere cierta información que no se encuentra en el sistema, la comunicación puede llevarse a cabo mediante cualquier método de interacción para objetos distribuidos, como CORBA o RMI de Java.
- Agente móvil: aquél que no está limitado al sistema donde se inició su ejecución, siendo capaz de transportarse de una máquina a otra a través de la red. Esta posibilidad le permite interactuar con el objeto deseado de forma directa sobre el sistema de agentes donde se encuentra dicho objeto. También puede utilizar los servicios ofrecidos por el sistema multiagente destinatario. Ningún sistema de objetos distribuido existente en la actualidad necesita utilizar agentes móviles para comunicarse.

Los agentes móviles suelen programarse normalmente en lenguajes interpretados o generadores de código intermedio –Telescript [URL10], Java [URL11], Tcl [URL12]–, ya que éstos dan un mejor soporte a entornos heterogéneos, permitiendo que los programas y sus datos sean independientes de la plataforma utilizada.

El proceso de transferencia de un agente de un sistema a otro suele realizarse en tres etapas [32]:

1. Iniciación de la transferencia.

- El agente identifica el destino deseado, realiza una petición de viaje al sistema y –si es aceptada– recibe el permiso para ejecutar la transferencia.
- El sistema suspende la ejecución del agente e identifica el estado y las partes del agente que serán enviadas.
- Se realiza la conversión en serie del código y del estado del agente (serialización) y se codifica según el protocolo establecido.
- El sistema hace la autenticación del agente.
- Se realiza la transferencia.

2. Recepción del agente.

- El sistema destinatario acredita al cliente.
- Se realiza la decodificación del agente y la conversión en serie del código y estado del agente (deserialización).
- El sistema crea la instancia del agente, restaura su estado y continúa la ejecución.

3. Transferencia de otras clases (sólo en sistemas orientados a objetos).

- Este proceso es necesario cuando el agente se mueve de un sistema a otro, cuando el agente se crea remotamente, o cuando necesita otros objetos. La transferencia de las clases puede realizarse completamente junto con el viaje del agente o hacer peticiones de carga cuando sea preciso.

En cuanto a la interoperación entre sistemas multiagentes cabe destacar que la normalización en el proceso de interconexión de agentes móviles suele aplicarse a dos niveles:

- Interoperación entre diferentes lenguajes de programación.
- Interoperación entre sistemas escritos en el mismo lenguaje.

La primera de ellas resulta muy compleja de alcanzar y continúa aún en proceso de estudio, mientras que el segundo caso está siendo normalizado por CORBA. No obstante, para alcanzar el grado de interconexión deseado, las MASIF de CORBA definen los siguientes conceptos destacables:

- Lugar: contexto dentro de un sistema donde puede ejecutarse un agente; por lo tanto, un agente viaja de un lugar a otro, ya sea en el mismo sistema o en otro distinto.
- Localización: va asociada con un lugar, indicando el nombre y la dirección que ocupa en el SMA.

- Localidad: propiedad de cercanía al destino, ya sea en el mismo ordenador o en la misma red.
- Infraestructura de comunicación: provee servicios de transporte al sistema.
- Autoridad: persona o entidad en nombre de la cual actúa un agente o un sistema de agentes.
- Región o dominio: conjunto de sistemas de agentes que tienen la misma autoridad y que no tienen que ser del mismo tipo.

Los agentes software móviles ofrecen una forma radicalmente diferente de controlar sistemas distribuidos con algunas ventajas clave como la simplicidad, robustez o el tamaño del código resultante. Los agentes móviles son un tipo de aplicaciones beneficiosas donde la robustez y la habilidad para la autoregulación son más importantes que el tiempo de respuesta. Pueden ser muy apropiados para llevar a cabo tareas de ajuste de parámetros en la gestión de sistemas. Este atributo permite a los agentes ejecutar tareas mundanas que sólo pueden ser realizadas por supervisores humanos.

Este tipo de agentes ha recibido una especial atención en los últimos años debido principalmente [10] a la proliferación de tecnologías para la integración de plataformas como Java RMI, CORBA [33], *middleware* y a la relativa disponibilidad de hardware potente. De esta forma se desea aportar nuevos mecanismos que permitan actualizar dinámicamente los procesos de control en redes de comunicaciones y en esta línea se han realizado dos importantes propuestas como es, por un lado, la tecnología de Redes Activas tal como presentaremos en el siguiente apartado, y, por otro, el uso de tecnología de agentes móviles que aporta la movilidad del código dentro de las redes.

Los agentes móviles se plantearon desde su inicio para aportar la movilidad del código ejecutable entre ordenadores. La idea básica es aprovechar los recursos de un ordenador cuando los de otro son escasos. Aunque intuitivamente la idea de la movilidad de los agentes parece bastante sencilla, la verdadera labor de investigación está en determinar la forma en que podemos mover la llamada inteligencia de los agentes de unos nodos de la red a otros.

En muchos casos quizás sea más eficiente el paso de mensajes tradicional que la utilización de código móvil de unos nodos a otros, sin embargo, son muy atractivas las posibilidades ofrecidas por los agentes móviles para aportar mecanismos útiles para situar dinámicamente nuevos controles software en entornos de redes. Los agentes móviles aportan la característica de no necesitar una conexión continua que conlleve el consumo de recursos de red. Además, este tipo de agentes pueden ser creados fuera de la red y cargados en la misma para realizar las tareas que tengan encomendadas por el usuario que puede invocar al agente cuando lo desee. Otra ventaja de los agentes móviles respecto al paso de mensajes se da cuando los ordenadores a comunicar están muy separados geográficamente que suele ser lo habitual en los entornos de comunicaciones. Además, en la mayor parte de situaciones, el problema no puede solucionarse únicamente con un paso de mensajes, sino que se requiere el envío de piezas de software, que mediante el uso de agentes es más ágil que a través del paso de mensajes. La literatura describe todas las ventajas aportadas por el uso de este tipo de agentes y la referencia [6] presenta una visión diferenciada entre agentes software y agentes móviles.

El uso de agentes móviles combina algunos de los beneficios del control centralizado y distribuido. En cada etapa de los algoritmos los agentes pueden aprender [34] sobre el crecimiento de la red interrogando a un nodo para descubrir su estado y el estado de los enlaces con sus vecinos. No es necesario almacenamiento de información centralizada porque no se mantiene información local en los agentes. Pueden fallar agentes individuales, mientras el sistema general continúa funcionando. Hay algoritmos que son muy complejos de implementar en una arquitectura completamente distribuida, sin embargo, son más simples si se aprovechan algunas de las ventajas del control distribuido.

Los agentes móviles aparentan ser distribuidos en el sentido que no hay controlador central. Sin embargo, los agentes tienen una visión del sistema distribuido que no es local y, desgraciadamente, puede ser global porque, en principio, un agente móvil podría visitar cada parte del sistema distribuido para capturar datos. Igual que los agentes tienen las ventajas de los sistemas centralizados, inherentemente padecen también los inconvenientes de estos sistemas. Las decisiones en el control centralizado pueden estar basados en datos anticuados, porque lleva cierto tiempo mantener todo el estado del sistema. Como los agentes deben obtener la información de las diferentes partes del sistema, puede ser que lo hagan de aquellas que aún no tienen los datos actualizados. El envejecimiento de los datos que usa un agente puede ser controlado restringiendo el número de nodos que un agente puede visitar para captar datos. Si el número de nodos visitados por un agente es pequeño, puede que sufra por un excesivo control general del sistema. Sin embargo, esta forma de actuación permite balancear las características entre control distribuido y centralizado.

Los agentes presentan comportamientos algorítmico y heurístico [35]. Tienen comportamiento algorítmico cuando calculan unas tablas de routing, y comportamiento heurístico cuando gestionan su carga de trabajo eligiendo qué agentes lanzar, terminar por sí mismos o moverse a otro nodo. Éste es un poderoso argumento para introducir un cierto grado de inteligencia en un sistema puramente algorítmico.

Los agentes software móviles son procesos computacionales con la capacidad del movimiento y basados en principios organizativos observados en insectos sociales como las hormigas. La aspiración es conseguir la emulación de esos principios para lograr trasladar a los sistemas de comunicaciones características de los sistemas naturales como la simplicidad, la adaptabilidad y la robustez.

A medida que crece la complejidad de los sistemas de telecomunicaciones, crecen también las debilidades del control completamente distribuido o centralizado. Los sistemas con control centralizado tienen problemas de escalabilidad con el tamaño del sistema, debido al incremento de comunicaciones y procesamiento, y son potencialmente vulnerables a fallos de control. En control distribuido, los procesos se ejecutan en procesadores diferentes. Los procesos distribuidos necesitan coordinarse entre sí para identificar que se ha producido un evento que requiere control. La decisión de la acción de control que hay que tomar es dependiente de las habilidades de los procesadores para cooperar.

Son necesarias nuevas formas para organizar los sistemas de control para aliviar su complejidad. Parece deseable que pudiesen unirse características de los sistemas centralizados y distribuidos en lugar de tener que decidirse por uno concreto. De este modo, necesitamos [35] control software que sea:

- Intrínsecamente robusto a los fallos del sistema y de programas.
- Intuitivo para ser escrito de una forma bien estructurada.
- Autoregulador.
- Requiera la mínima cooperación directa entre procesos.

Las características anteriores, requeridas para los sistemas de control, se dan desde hace años en las hormigas. Una colonia de hormigas es claramente un sistema distribuido. Sin embargo, las hormigas son móviles, todo lo contrario a los agentes de un sistema distribuido convencional. Observando la forma de localización de comida en las colonias de hormigas se observó:

- Las hormigas son móviles.
- Los mensajes que intercambian entre sí son estacionarios.
- Los mensajes no son dirigidos a una hormiga específica.
- Los mensajes se atenúan con el paso del tiempo.
- Algunos de sus comportamientos son aleatorios.
- Existen un gran número de sencillas hormigas en lugar de unas cuantas complejas.

Precisamente las anteriores características forman la base de la mayoría de los agentes móviles. La movilidad es conseguida permitiendo a los agentes transmitirse para continuar trabajando en otros procesadores del sistema donde se desarrollan. Cada vez que un agente se mueve a otro procesador diferente deja un mensaje estático que puede ser leído por otro agente que venga detrás de él. Esto permite que otros agentes puedan trabajar adecuadamente. Éste es un procedimiento muy adecuado para sustituir agentes erróneos por nuevos agentes para compartir una tarea cuando los recursos son insuficientes. La atenuación de los mensajes se consigue con un *time-stamp* que va haciendo que los mensajes sean poco a poco menos relevantes. De este modo, los mensajes erróneos, o que han perdido importancia, pueden ser sustituidos del sistema. El comportamiento aleatorio también puede ser introducido en los agentes. En un sistema realmente complejo el número de excepciones que necesita un agente para cubrir cada situación puede ser inabordable, por lo que puede ser de interés permitir a los agentes solventar situaciones mediante elecciones aleatorias que no tienen por qué ser previamente preprogramadas en los roles del agente. Por último, si podemos diseñar un sistema de control de agentes sencillos, éste será más pequeño, más fiable y fácil de programar.

El trabajo [35] presenta un sistema de control basado en un agente móvil, que cumple con una disciplina de programación que está bien estructurada y es robusta ante el fallo de un agente o de un componente del sistema distribuido. Para ello se proponen las tres siguientes características en los agentes para que sean intrínsecamente robustos:

- No debería haber intercomunicación directa entre agentes. Si no existe esa intercomunicación, un agente no podría enviar un mensaje cuyo destino sea una instancia específica de otro agente. Por tanto,

un agente no debería enviar ni recibir un mensaje desde una instancia específica de otro agente. Esto aporta a los agentes móviles autonomía e independencia, que es una importante característica para la robustez. Sin embargo, los agentes deben disponer de algún mecanismo para que puedan pasarse los resultados de sus actividades entre sí. La forma de hacer esto es a través de la modificación del estado del sistema que controlan los agentes. Este tipo de comunicación sólo es posible si los agentes son móviles.

- Los agentes deberían ser un número razonablemente alto. De este modo, a lo que se aspira es a que entre pocos agentes soporten una importante proporción de la responsabilidad de control del sistema. La reunión de agentes será intrínsecamente robusta al fallo de un agente individual. Por supuesto, el fallo de un agente causará una cierta degradación del sistema, pero será gradual en lugar de catastrófica. La gradual, en lugar de repentina, degradación del rendimiento es una de las ventajas más significativas del uso de agentes móviles.
- Los agentes deberían ser capaces de alterar dinámicamente sus tareas de asignación y su número. Esta tercera propiedad implica que el número de agentes debería ser proporcional a la carga de trabajo y que los agentes que fallen podrán ser detectados y reemplazados. Los agentes fallidos deben ser detectados por el conjunto de agentes que deberán compensar los errores incrementando el número de agentes al nivel previo al fallo.

El mecanismo propuesto para lograr lo anterior es el uso de *time-stamp* en los elementos de datos que son modificados por los agentes como parte de su función normal cuando visitan un nodo. Éste es un dispositivo muy sencillo pero con un potencial importante para permitir la unión de los agentes para su auto-organización. Los agentes cuando se mueven por la red pueden juzgar si una tarea tiene o no recursos mediante la edad de los mensajes encontrados. Es decir, si un agente encuentra mensajes recientes relativos a su tarea actual, podrá asumir que la tarea estará siendo realizada por otro agente.

El flujo de tráfico en la red es simulado por un módulo generador de tráfico. Un perfil de tráfico determina los nodos fuente y destino, tiempo de inicio, duración y ancho de banda de las conexiones que van a añadirse a la red. Este control de sistemas de agentes móviles puede ser entendido como un ecosistema de entidades de control. Para añadir nuevas funciones de control, puede crearse un nuevo tipo de agente, dando mecanismos para interactuar y gestionar su población y después crear el ecosistema. Esto hace el sistema altamente extensible.

El artículo [36] presenta el uso de agentes móviles para distribuir y automatizar las funciones de gestión de red en ATM. La gestión a través de agentes móviles ofrece un paradigma para la gestión de red distribuida, flexible, escalable y robusta que mejora las limitaciones de los actuales esquemas de gestión centralizada. Los sistemas de gestión de red se basan en un paradigma trabajo-intensivo con plataforma de inteligencia centralizada y aplicada a datos distribuidos. Los datos operativos son capturados por agentes empotrados en dispositivos de red y recogidos por una plataforma centralizada usando un protocolo de gestión como SNMP o CMIP. La centralización presenta conocidas limitaciones como la escalabilidad y la adaptabilidad de la gestión. Es decir, cuando el tamaño (número de dispositivos), complejidad (número de variables gestionadas), o la velocidad de la red crece, o si las velocidades de gestión de la red son limitadas, el sistema de gestión centralizada se convierte rápidamente en inmanejable. Para solventar estos y otros problemas de la gestión centralizada se han propuesto múltiples soluciones descentralizadas [37-40]. Es importante comentar también que la gestión descentralizada ha sido seguida de lo que se ha denominado gestión por delegación [41], que aprovecha la movilidad del código.

La tecnología de código móvil suele dividirse en código-bajo-demanda, evaluación remota y agentes móviles [42]. Mientras en código-bajo-demanda y evaluación remota la atención se fija en la transferencia de código entre componentes, en el paradigma agente móvil se mueve a un lugar remoto un componente computacional completo junto con su estado (el código que necesita) y algunos de los recursos necesarios para realizar las tareas a desempeñar.

El artículo [36] presenta una propuesta de sistema de gestión para conmutadores ATM con un paradigma basado en agentes móviles que implementan las funciones de gestión de fallos y rendimiento. Como la gestión de fallos puede ocurrir en los periodos duros de congestión, la función de gestión es bastante importante, aunque la función de gestión de rendimiento pueda sobrecargar el tráfico de la red de forma importante con respecto a otras funciones. Por esta razón estas labores se realizan esporádicamente para no sobrecargar innecesariamente la red.

6.9.2. GESTIÓN DISTRIBUIDA DE FALLOS EN REDES ATM MEDIANTE AGENTES

En la referencia [43] se presentan varios problemas que son solventados mediante agentes. Este trabajo demuestra que la comunicación entre los agentes es, o puede ser, el verdadero cuello de botella de esta propuesta que obtiene muy buenos resultados en aplicaciones sobre la gestión de fallos sobre redes locales ATM. Este trabajo propone el uso de agentes mediante un sistema distribuido dividido en tres grandes módulos: las fuentes de conocimiento, el módulo caja-negra y el de mecanismos de control. Es destacable que esta propuesta propone una arquitectura con un paradigma distribuido pero situado en capas superiores de la tecnología ATM. En realidad, la propuesta está situada por encima de la capa AAL-5, lo que acaba redundando en limitaciones en su rendimiento.

6.9.3. GESTIÓN DE REDES ATM MEDIANTE SMA

La referencia [6] describe la metodología software adoptada para la construcción de un sistema de gestión de red basado en un sistema multiagente. El proyecto Tele-MACS [URL13] es una aproximación al problema de la configuración (asignación de rutas y de ancho de banda) lógica de recursos de red dinámicamente adaptada a la utilización de los usuarios de la misma. Esto se consigue con técnicas de control basadas en el uso de un SMA distribuido. Se presenta la arquitectura de control multiagente distribuido. Se acoplan las características de planificación de los tipos de agentes con las actividades de agentes reactivos, para que el sistema de gestión de red pueda adaptar la configuración de recursos para cumplir con los cambios en la solicitud de conexión de los usuarios.

En las redes ATM la importancia del control de red fue situada inicialmente en el desarrollo de algoritmos que pudiesen estimar la utilización del ancho de banda para ciertos tipos de conexión a elevados grados de probabilidad. Una razón para esto fue que, en sistemas de comunicación de alta velocidad, las estrategias de control trabajasen en escalas de tiempo más pequeñas que las del plano de gestión. Sin embargo, en redes reales, estas funciones de control no actúan independientemente, pero son parte de una estrategia general. De este modo, la aspiración de la gestión de recursos de red es ofrecer al cliente la QoS requerida, a la vez que se permite al operador de la red usar ésta de una forma económica y eficiente. Aunque ATM fue concebida como una tecnología para satisfacer la reconfiguración dinámica de recursos de red en tiempo de ejecución, uno de los mayores problemas es la complejidad resultante. Una de las formas de solucionar este problema es distribuir el control y permitirle a agentes especializados la gestión de tareas concretas. Otra forma de reducir la complejidad resultante ha sido el control jerárquico y la planificación de comportamiento pro-activo. Tele-MACS explota la potencia de ambos métodos.

Considerando la unión entre planificación y reacción en los sistemas de control, se ha desarrollado un entorno de agentes multi-nivel que contempla el aumento de características de agentes de planificación y reactivos, cooperando en un mundo común. La idea está basada en la construcción de un sistema de gestión inteligente desde múltiples capas. Tele-MACS se basa en capas porque permite aislar o encapsular los agentes interactuando con un cierto entorno que aporta extensibilidad (creando interfaces bien definidas entre las capas) y que ofrece robustez a los fallos de software y robustez contra la imposibilidad de realizar una secuencia de acciones en el intervalo de tiempo del ciclo de control (las capas permiten operar en diferentes escalas de tiempo). Primero se construye el sistema coordinado completo que consigue el propósito específico a un cierto grado de competencias. Después se testea el sistema y se realizan ajustes hasta que se considera que el sistema satisface los objetivos previstos. Finalmente se construye otra capa completa de control a alto nivel de competencia y se ajusta a un mecanismo de supresión.

Algunas de las ventajas del uso de procedimientos tradicionales de control de admisión de conexión (CAC) son:

- En general, el algoritmo de CAC debe ser ejecutado en cada conmutador de una ruta candidata. Esto es, cada conmutador individual determina si puede soportar o no la llamada en un modo local. Esto implica un cierto *overhead*, especialmente en casos en que la función de conexión sea rechazada en alguna ocasión a lo largo de toda la ruta.
- La mayor parte de los algoritmos CAC incorporan simplificaciones, y no se pueden usar para el rango completo de servicios que pueden ser soportados por las redes ATM, ya sea en la actualidad o en el futuro.
- Varios algoritmos CAC eficientes, que intentan explotar al máximo las ventajas de la multiplexación, no pueden ser implementadas en tiempo real.

Son conocidos los variados problemas que se encuentran en la construcción de sistemas software para la resolución de problemas grandes y complejos. El proceso completo de diseño, modelado e implementación

requiere un importante esfuerzo, tiempo y dinero. Aparecen así las ventajas conceptuales ofrecidas por la programación orientada a objetos (POO) que son bien conocidas; sin embargo, el diseño de SMA puede ser visto como una extensión del diseño OO, especialmente cuando muchos investigadores consideran el diseño de SMA como una metáfora más abstracta.

En Tele-MACS primero se describen las bases de cada agente y algunas de las funciones que realizan. Después se ofrecen detalles de sus interacciones en el contexto de control de comunicaciones.

Se describe una demostración intuitiva de cómo se ha usado la idea de crear un sistema de gestión multiagente que pueda controlar selección de rutas de ancho de banda dinámico en redes ATM para evitar las congestiones. Un agente de CAC (CAC Agent) decide sobre las siguientes opciones:

- Hay suficiente ancho de banda disponible y la conexión puede ser aceptada.
- El ancho de banda es insuficiente, pero la conexión puede ser aceptada si se modifican los VPs usados.
- Hay insuficiente ancho de banda, pero puede negociarse con otro proveedor de servicio para obtener más.
- No se podrá obtener el ancho de banda y la solicitud de conexión es rechazada.

Hay dos tipos de agentes de usuario: el PUA (Proxy User Agent) y los PCA (Proxy Connection Agents) que tienen una funcionalidad similar. La diferencia es que el PUA de flujo ascendente conecta a una aplicación llamante y el PUA de flujo descendente a una aplicación llamada, mientras el PCA es la interfaz a solicitud de conexiones ascendente o descendente desde otra red. El PUA colecciona los parámetros de los mensajes de señalización y después comunica con el PUA destino para investigar cuándo el sistema final destino acepta la solicitud de conexión. El PUA envía después un mensaje al CACA del nodo fuente negociando los parámetros de la solicitud de conexión. El Agente CAC interroga a los posibles RA (Resource Agents) que sean capaces de ofrecer los requerimientos de recursos específicos. El Agente CAC toma las respuestas y decide el RA preferido para establecer la conexión. El RA elegido interactúa con los agentes Switch Wrapper (SwWA) para hacer las conexiones cruzadas en cada conmutador a través del *path* seleccionado. Los SwWA implementan una API que mapea los mensajes de los agentes a los protocolo de señalización ATM. El agente SwWA constituye una abstracción software virtual de un conmutador ATM y sus recursos, y ofrece una interfaz genérica independiente del vendedor para control de red y aplicaciones de administración.

6.10. REDES ACTIVAS

Desde sus inicios, las redes han dispuesto de una gran variedad de elementos hardware (*switches, routers, bridges, brouters, hubs*, sistemas finales, etc.) con funciones bien conocidas (conmutación, enrutamiento, puenteo, control de congestión y de flujo, garantía de QoS, ejecución de aplicaciones, etc.). Las redes actuales son canales de comunicación que transfieren paquetes entre sistemas finales usando el hardware citado anteriormente. Pero también existen nuevas investigaciones para equipar los elementos hardware con elevadas prestaciones mediante técnicas software. Esto permite equipar la red con características activas (*activenets*⁸) en las que los elementos hardware computan, cambian y operan los paquetes y también transfieren y propagan código. Así, una red activa es una red programable que permite que el código sea cargado dinámicamente en los nodos de la red en tiempo de ejecución. La literatura sobre redes activas [28,44-57] estudia varios mecanismos para aprovechar los nodos activos. Sin embargo, las propuestas son insuficientes para las redes de tecnología ATM y la referencia [28] es un ejemplo de estas investigaciones para ATM.

Las redes IP basadas en protocolos estandarizados *de facto* se encuentran con la situación de tener que operar sobre muy diversas redes físicas actuando a modo de capa de red virtual encima de estas redes. Ante esto se realizan propuestas para conseguir redes más adaptables, tanto a la red física, como a las aplicaciones que se estén usando sobre ellas. Como IP no dispone de la habilidad de adaptarse a las necesidades de cambios, en las aplicaciones se propuso [45] un nuevo tipo de arquitectura de red llamada red activa. Esta arquitectura permite a las aplicaciones extender sus funcionalidades de una forma dinámica inyectando protocolos personalizados, también conocidos como *protocolos específicos de las aplicaciones*. En este tipo de redes los paquete eligen el protocolo por el que desean ser procesados.

⁸ La noción Active Network representa la idea que los nodos de las redes de comunicación pasan a ser activos por tomar parte en el procesamiento de las aplicaciones y en el soporte de servicios personalizados.

No hay consenso para decidir cuándo una red es activa, pero existen dos grandes tendencias: "...una red es activa si incorpora nodos activos con la capacidad de ejecutar programas de usuario." , o bien si ésta "...implementa mecanismos de propagación de código entre los dispositivos de interconexión (conmutadores ATM en nuestro caso)". La arquitectura TAP es activa en ambos sentidos porque sitúa nodos activos en puntos estratégicos que implementan un protocolo activo que permite que el código de usuario sea cargado dinámicamente en los nodos de la red en tiempo de ejecución. También se ofrece el soporte de propagación de código en la red gracias a las células RM. TAP es también una arquitectura distribuida en el sentido que el protocolo usa varios agentes coordinados y auto-colaborativos entre conmutadores activos.

Las redes activas, abiertas y programables son por tanto una nueva área técnica [44-57] para explorar vías por las que los elementos de la red puedan ser dinámicamente reprogramados por administradores, operadores o usuarios generales para obtener la QoS requerida y otras características, como servicios personalizados. Esto ofrece atractivas ventajas y también cambios importantes en aspectos como el rendimiento, la seguridad y la fiabilidad. Por tanto, ésta es una línea abierta para la investigación y el desarrollo de la ingeniería de protocolos que permita mover el código de servicio (colocado dentro de la capa de transporte de red) a los nodos de conmutación. La bibliografía en este campo estudia varios mecanismos para obtener ventaja de los nodos activos. Algunas de las ventajas de los protocolos activos son conseguidas instalando los nodos activos en puntos estratégicos de la red. Conceptos como redes activas, protocolos *boosters* o agentes software fueron propuestos y desarrollados para redes IP; sin embargo, como hemos dicho antes, las propuestas son insuficientes en el ámbito de las redes de tecnología ATM que es donde hemos situado esta tesis.

Las redes activas proponen otro novedoso paradigma para la construcción de arquitecturas de red basadas en las novedades introducidas por protocolos de comunicaciones como IPv6. Este paradigma se centra en los routers y conmutadores que forman la red, donde los nodos aportan la posibilidad de aplicar acciones de usuarios sobre los paquetes de datos que fluyen por la red. Este planteamiento es parecido a la tecnología basada en agentes móviles, donde los paquetes que fluyen por la red puede ser vistos como pequeños programas que pueden ser ejecutados por los routers y conmutadores. Una distinción [10] que puede destacarse entre los agentes móviles y los pequeños paquetes de código (aunque en muchas investigaciones se suele denominar indistintamente al código móvil y a los agentes móviles) ejecutable es la habilidad de los agentes móviles para retener el estado de un punto de ejecución al siguiente punto.

En la visión integrada de las redes activas los paquetes transmitidos (cápsulas) contienen, además de los datos, fragmentos de programas responsables del procesamiento de los propios datos en los conmutadores. No obstante, también existe un planteamiento discreto en las *active-nets* donde la organización del servicio es realizada separadamente del propio procesamiento del servicio, es decir, lo que conocemos como *out-band*. Se consideran como redes activas las redes de conmutación de paquetes (IP en su gran mayoría) cuyos paquetes pueden contener fragmentos de código para ser ejecutado en los nodos que forman la red. Los nodos extremos de la comunicación se encargan de introducir o inyectar código en la red para cambiar el comportamiento de ésta según sus necesidades. Pero esta inyección de código en la red ha dado lugar a dos corrientes bastante claras en la investigación de las redes activas:

- Conmutadores programables [47,48], pensados para la inyección del código por parte del operador de la red. Ésta es la anteriormente comentada como *out-band*, donde el código es insertado en la red a través de los conmutadores programables preparados al efecto.
- Las cápsulas [49] aportan la visión de que los paquetes de datos transportan pequeños trozos de código de programas que son transportados *in-band* y ejecutados en cada nodo que es encontrado en el camino de la comunicación. Esta visión es la que realmente introduce un nuevo paradigma en las redes de paquetes, que dejan de ser pasivas para ser activas al ser capaces, no sólo de reenviar, enrutar o conmutar paquetes, sino también de ejecutar el código que transportan los paquetes.

Aunque los planteamientos de estas dos ideas son bastante diferentes, las posibilidades de convergencia entre el concepto de red puramente programable y el soporte de cápsulas empiezan a aparecer como factible. Esa posibilidad de coincidencia está en el hecho que hacer programable un nodo requiere más bien de programación especializada que posiblemente el operador de la red no desee realizar y, por otro lado, la generación de cápsulas puede dejarse a la intervención de los operadores de la red. Las cápsulas son más propiamente específicas de las aplicaciones que específicas de los usuarios, por lo que, en realidad pueden desempeñar la función propia del código que los operadores deseen usar al iniciar ciertas aplicaciones.

ANTS (Active Node Transport System) [50] es una de las implementaciones de referencia en el campo de las redes activas. Emplea una variante de la visión *in-band* para construir su arquitectura. En lugar de emplear el transporte constante de código con cada paquete, ANTS almacena el código de usuario más reciente para evitar recargar el código en cada grupo de paquetes que son llamados cápsulas. Las cápsulas

transportan los valores de los parámetros de cada pieza de código. Si el nodo a través del que pasa un paquete contiene el código relacionado, el nodo inicializa el código con los valores de los parámetros de ese paquete y lo ejecuta. Si el código no está presente en ese nodo, el nodo solicita el código a su vecino más cercano en el sentido contrario al del flujo de datos. Hemos de destacar que en cierto sentido hemos inspirado nuestra propuesta en este tipo de transporte de código pero debidamente ajustado a un escenario completamente diferente como es el de la arquitectura TAP.

En realidad, la visión que nosotros damos al protocolo activo está apoyada en las dos vertientes del concepto de redes activas. Por un lado, la existencia de los agentes programables que usamos está relacionada con la ejecución o elección de una CoS concreta. Para nosotros las cápsulas son las células RM que se generan en las retransmisiones que, en alguna de las variantes de nuestra propuesta pueden transportar código muy simple para realizar la búsqueda de las PDUs que hay que retransmitir. Por otro lado, la posibilidad de intervención del operador de la red sobre los agentes programables aporta la vertiente *out-band* en la red activa.

6.11. SISTEMA MULTIAGENTE TAP

Una vez presentados los conceptos principales sobre los que se apoya la tecnología de agentes software, pasamos a describir resumidamente nuestra propuesta, destacando en primer lugar su área de aplicación y clasificación, para comentar después la arquitectura completa del SMA-TAP, así como los agentes que conforman la arquitectura y sus flujos, cuya evolución ha sido propuesta en [58,59].

Comenzamos destacando que la arquitectura TAP está basada en la incorporación de varios agentes software en el modelo de conmutador propuesto y que hemos denominado AcTMs (conmutadores ATM activos). De este modo damos lugar a una red ATM activa en la que sus conmutadores van a desempeñar una función activa en la gestión de las colas de entrada, en la detección de congestiones, así como en la labor de retransmisiones cuando empiezan a perderse células de las fuentes de tráfico que se han elegido con tráfico garantizado.

La *Figura 6.4* permite observar la posición que ocupa el SMA-TAP dentro del modelo arquitectónico ATM. Como puede comprobarse, el SMA forma parte del Plano de Gestión (superficie rayada) ya que aporta nuevos mecanismos para la gestión de recursos ATM, siendo transparente a funciones como CAC y UPC. Pueden observarse también en la *Figura 6.4* las superficies rayadas donde se incorpora nuestra propuesta de protocolo TAP del que forma parte la extensión EAAL-5 que hemos desarrollado.

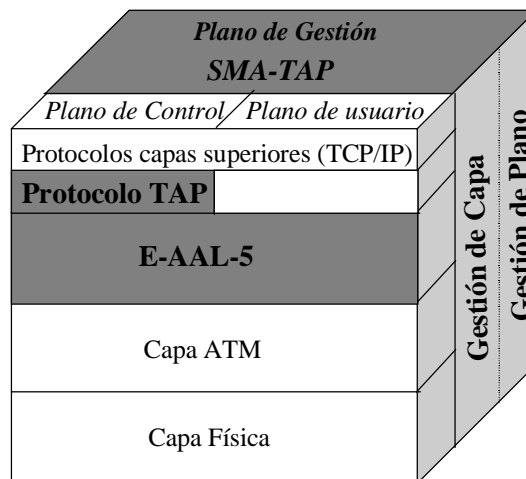


Figura 6.4. Situación del SMA-TAP en el modelo arquitectónico ATM

En el caso de TAP no pretendemos enfrentarnos a escenarios de control de tráfico ni de administración de red, ya que estas tareas ya han sido fuente de múltiples investigaciones, aunque no demasiadas en el ámbito de los SMA. Las funciones CAC y UPC son claras candidatas a ser implementadas mediante agentes software en los que se delegan las labores de establecimiento de la conexión, el control de flujo y la resolución de situaciones de congestión de forma reactiva. Del mismo modo, la mayor parte de investigaciones relacionadas con la aplicación de SMA a ATM están relacionadas con el mantenimiento de los parámetros de QoS, el re-enrutamiento y el tráfico cuando un determinado enlace experimenta problemas.

En la mayor parte de las propuestas, los esfuerzos se han centrado en conseguir el mayor throughput posible. En muchos casos se asume que la QoS es uno de los factores de utilidad más importantes de las redes ATM, pero la maximización del throughput del tráfico en las redes de alta velocidad no es necesariamente el parámetro más valioso para los usuarios de la red, y de hecho, desde el punto de vista de un operador de red ATM, es más importante mantener el servicio dentro de unos márgenes de satisfacción de los usuarios de la red que el de intentar buscar la optimización del ancho de banda arriesgándose a la pérdida de la GoS. Por esto TAP centra sus objetivos en la consecución de la garantía de servicio, aunque es claro que este SMA puede ser fácilmente ampliado con nuevas funcionalidades más propias de la capa de control del modelo de referencia ATM (*Figura 6.4*).

6.11.1. ARQUITECTURA DEL SMA-TAP

En esta sección presentamos el esquema general de la arquitectura TAP, donde podemos establecer un subsistema constituido por el SMA-TAP que está formado por cinco agentes software.

La *Figura 6.5* presenta la arquitectura TAP completa, incluyendo la memoria DMTE, las colas de entrada, el buffer, las tablas de E/S, así como los agentes software que forman el subsistema SMA-TAP. Las fuentes garantizadas generan su flujo que llega a la DMTE y es procesado a los puertos de salida que multiplexan las células hasta el siguiente conmutador activo.

A continuación comentamos brevemente la funcionalidad de cada uno de los agentes software que serán estudiados detalladamente en el *Capítulo 10*. Los diversos agentes software que constituyen el SMA-TAP son los siguientes:

- El agente CoSA (Class of Service Agent), como podemos observar en la *Figura 6.5*, es un agente programable que se encarga de recibir los parámetros que caracterizan el tráfico que proviene de las fuentes de entrada. En nuestro caso, este agente se encarga del establecimiento de la conexión, de las tablas de routing para determinar los VPI/VCI extremo-extremo y también de la liberación de la conexión.
- El agente WFQA (Weighted Fair Queueing Agent) mantiene una coordinación constante con el agente CoSA, de forma que recibe de éste los flujos de entrada destinados a las colas que WFQA gestiona de forma justa y de acuerdo a los pesos establecidos en los parámetros de tráfico.
- El agente programable CCA (Control Congestion Agent) controla las congestiones basándose en EPDR y en otros algoritmos que el administrador de la red puede elegir. Este agente monitoriza el buffer de entrada, y cuando su ocupación sobrepasa el umbral establecido no acepta ninguna PDU entrante. Del mismo modo, CCA puede ser considerado como un agente inteligente ya que es capaz de autoajustar el valor umbral del buffer en función del tráfico que se está produciendo. Como ya hemos visto en el *Capítulo 5*, la elección del umbral es una labor compleja que está en función del tráfico generado por las fuentes.
- El agente CCA se coordina con el agente RCA para solicitar las PDU al conmutador activo anterior. La función del agente RCA es generar células RM nativas que son transmitidas en sentido contrario al flujo de información. Los conmutadores no activos reconocen las células RM como células RM de TAP y no toman ninguna otra acción que reenviarlas hacia la fuente emisora de los datos.
- El agente DPA se encarga de obtener las PDU o células del buffer de entrada cuando son transmisiones, o PDU de la DMTE cuando se trata de retransmisiones. Una vez recibidas, se encargará de despacharlas completas al correspondiente puerto de salida, previa actualización de la Tabla de Entrada/Salida de ese puerto. Con esta técnica se evitan las congestiones de los puertos de salida y también la mezcla de células pertenecientes a diferentes PDU o fuentes.

6.11.2. CLASIFICACIÓN DEL SMA-TAP

En apartados anteriores se ha realizado la revisión de la literatura en materia de agentes, centrándonos en los aspectos más interesantes relativos a los sistemas de telecomunicaciones. Una vez hecho este ejercicio y, ya presentada la arquitectura general del SMA sobre los conmutadores ActMs, pasaremos a realizar una clasificación de SMA-TAP atendiendo a la funcionalidad de los agentes que intervienen en el sistema. Antes de ello queremos presentar la forma en que se mapean los agentes en las capas del modelo de referencia ATM. La *Figura 6.6* muestra cómo el agente CoSA está situado en la capa de Control de Plano, mientras los cuatro agentes restantes forman parte de la Capa de Gestión de Planos. Podemos observar también las vías de comunicación entre cada uno de los agentes.

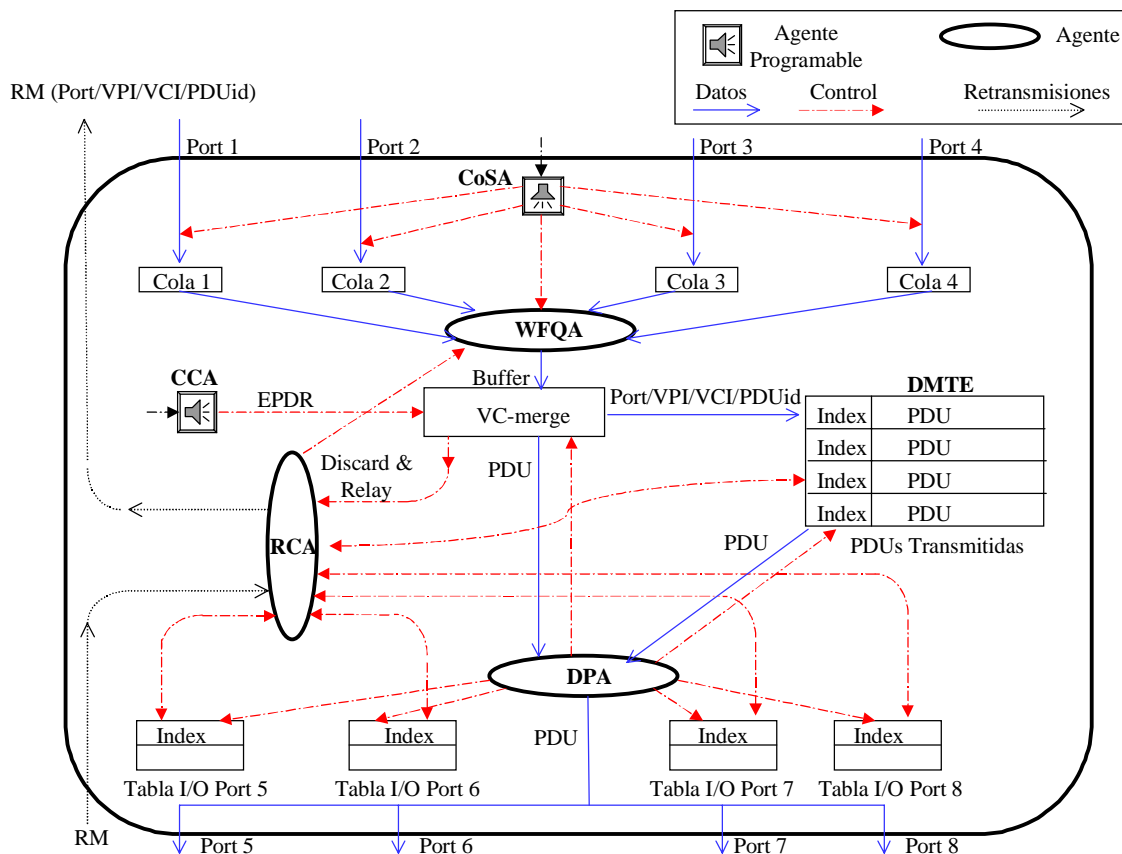


Figura 6.5. Arquitectura TAP con el subsistema SMA-TAP

Estamos por tanto ante un sistema multiagente constituido por un grupo de cinco agentes que colaboran conjuntamente para resolver muy diversas tareas. Desde el punto de vista topológico nuestro sistema no se ajusta plenamente a ninguno de los dos planteamientos generales comentados anteriormente. Es decir, no estamos ante un esquema completamente mallado porque no todos los nodos de nuestra red VPN necesitan tener instalado el SMA-TAP para poder disponer de las prestaciones que aporta la arquitectura completa. Por tanto no se requiere la conectividad total de todos los agentes con todos los nodos, lo que reduce sustancialmente, tanto el nivel de complejidad, como la robustez y la escalabilidad del sistema. Tampoco estamos ante una topología jerárquica en el sentido clásico de la palabra, aunque en realidad TAP responde plenamente a este esquema, sólo que de una forma más completa que la presentada en la *Figura 6.4*. Veremos en el Capítulo 7 que estamos ante un sistema claramente distribuido en la red a través de una topología arborescente que soporta las conexiones punto-punto, punto-multipunto y multipunto-multipunto.

Desde el punto de vista funcional de los agentes disponemos de un agente programable, que puede actuar como agente inteligente y también como agente reactivo ante situaciones de congestión. El agente CCA no sólo permite al operador elegir el algoritmo y los parámetros del esquema de control de congestión, si no que se comporta reactivamente ante la aparición de congestiones, y proactivamente pudiendo actuar para evitar la aparición de congestiones ajustando el valor umbral del buffer. Por otro lado, WFQA es un agente colaborativo que recibe eventos desde los agentes CoSA y RCA para llevar a cabo sus acciones generando como salida células y/o PDU clasificadas de acuerdo a los pesos de sus fuentes. El agente CoSA, también programable y reactivo, responde ante el establecimiento de conexión de las fuentes, para lo que necesita coordinarse con otros agentes similares en conmutadores adyacentes. El agente RCA, también reactivo, es capaz de atender eventos provocados por situaciones de congestión en su propio conmutador y en otros conmutadores de la red. Este agente colabora también directamente con el agente WFQA y de forma indirecta con CCA. RCA podría ser considerado como un agente móvil, no en el sentido de los agentes, sino según el punto de vista de las redes activas, porque permite la movilidad de código sencillo entre conmutadores a través de las células RM. Por último, DPA es un agente autónomo que, mediante técnicas de control, se encarga de actualizar las estructuras de datos del sistema como las Tablas de E/S, la memoria DMTE y el buffer del conmutador.

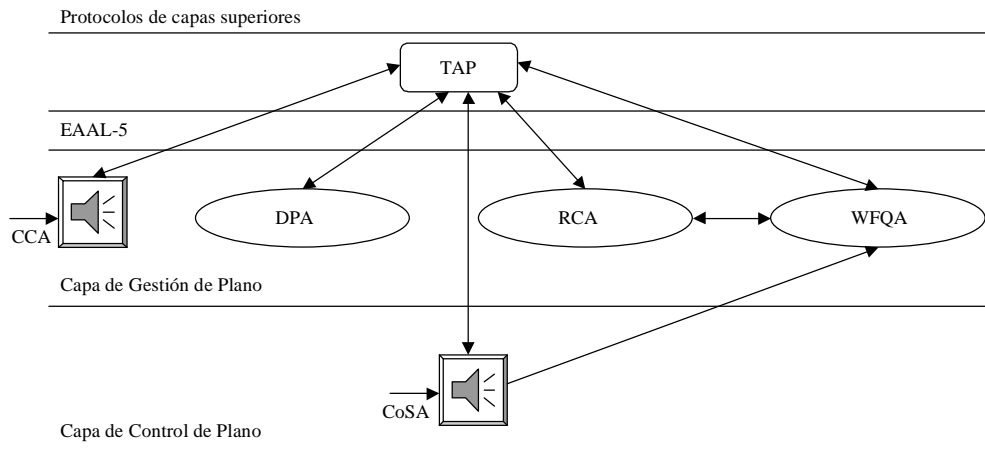


Figura 6.6. Arquitectura SMA-TAP en capas

6.12. CONCLUSIONES

Desde luego, no es evidente que la utilización de redes programables, por cualquiera de los métodos comentados, sea claramente útil ya que supone que hay que pagar ciertos costes. Tanto las ventajas como inconvenientes pueden estudiarse desde tres puntos de vista: impacto en los servicios de la red, impacto en el rendimiento de la propia red y facilidad en el desarrollo de protocolos. Existen múltiples investigaciones que demuestran tanto la utilidad de la inclusión de agentes software en la red, como el desarrollo de redes activas, y la mayor parte de ellas acaban beneficiando a uno o todos los puntos de vista anteriores. Este capítulo ha aclarado conceptos y presentado una taxonomía del ámbito de las redes programables, donde podemos comprobar la amplitud de horizontes planteados en las investigaciones actuales. Estas investigaciones indagan en la introducción e integración de agentes programables y agentes móviles en las arquitecturas emergentes de telecomunicaciones. El concepto general de agentes aporta un nuevo paradigma por la rápida y parametrizable provisión de servicios en entornos de procesamiento distribuido. Las investigaciones presentadas proponen mejoras para solventar ciertas limitaciones de las propuestas estándares y aprovechamos para incluir en la tecnología ATM un importante objetivo final como es la visión de redes ATM activas o redes programables abiertas (open programmable networks).

Uniendo la visión de los SMA y de las redes activas hemos justificado la propuesta de nuestro SMA-TAP que se basa en una arquitectura de cinco agentes software que soportan el protocolo TAP sobre los conmutadores activos AcTMs inspirados en algunos de los fundamentos de las redes activas. De este modo nos encontramos ante un protocolo activo capaz de aportar la GoS deseada a las conexiones que el usuario o el operador de la red decidan. El protocolo TAP que proponemos es activo desde los dos puntos de vista de las redes activas: porque la arquitectura que lo soporta dispone de nodos activos en puntos estratégicos de la red que implementan un protocolo activo que permite que el código de usuario sea cargado en los nodos de la red en tiempo de ejecución, y porque, además, la arquitectura y el protocolo soportarían la propagación de código gracias al uso de células RM que se pueden encargar de transportar tanto código como datos.

REFERENCIAS

- [1] *Diccionario de la Lengua Española. Real Academia Española. XXI Edición, (1992).*
- [2] *Webster's Encyclopedic Unabridged Dictionary of the English Language. Ed. Gramercy (1996).*
- [3] Object Management Group, "Agent Technology Green paper," *OMG Document ec/99-12-02, (1999).*
- [4] S. Franklin, A. Graesser. "Is it an Agent or just a Program?: A Taxonomy for Autonomous Agents," *Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, (1996).*
- [5] Wooldridge, M. and Jennings, N. R., "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review. Vol. 10. No. 2, (1995).*
- [6] Alex L. G. Hayzelden and John Bigham (Eds.), "Software Agents for Future Communications Systems," *Springer, (1999).*
- [7] David D. Clark and David L. Tennenhouse, "Architertural considerations for a new generation of protocols," *ACM SIGCOMM, (1990).*

- [8] Nwana, H. S., Azarmi, N. (Eds.) "Software agents and Soft Computing: Towards Enhancing Machine Intelligence," *Springer-Verlag*, (1997).
- [9] Müller, J. P., "Architecture and Applications of Intelligent Agents: A survey," *The Knowledge Engineering Review*, Vol. 13. No. 4, (1998).
- [10] Alex L. G. Hayzelden and John Bigham, "Agent Technology in Communications Systems: An Overview," <http://www.agentcom.org/agentcom/>, (1999).
- [11] Green, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F., Evans, R. "Software Agents: A review," *Technical Report of Trinity College. University of Dublin*. http://www.cs.tcd.ie/research_groups/aig/aig/, (1997).
- [12] Magedanz, T., Rothermel, K., Krause, S., "Intelligent Agents: An emerging Technology Next Generation Telecommunications?," *INFOCOM'96*, (1996).
- [13] Russell, S. Norvig, P., "Artificial Intelligence: A modern Approach," *Prentice Hall*, (1995).
- [14] H. S. Nwana. "Software Agents: An Overview," *Knowledge Engineering Review*, (1996).
- [15] N. R. Jennings, K. Sycara, M. Wooldridge. "A Roadmap of Agent Research and Development," *Autonomous Agents and Multi-Agent Systems*, I, (1998).
- [16] M. P. Singh. "Agent Communication Languages: Rethinking the Principles," *IEEE Computer Vol. 31*, (1998).
- [17] Pere Vilà and Josep L. Marzo, "Scalability Study and Distributed Simulations of an ATM Network Management System based on Intelligent Agents," *SPECTS'2K 2000 SCS Symposium on Performance Evaluation*, (2000).
- [18] L.C. Lee, H. S. Nwana, D. T. Ndumu and P. De Wilde, "The stability, scalability and performance of multi-agent systems," <http://www.labs.bt.com/projects/agents/publish/papers.htm>, (1998).
- [19] Mark Greaves, Heather Holmback and Jeffrey Bradshaw, "What Is a Conversation Policy?," *KAW'99 Proc. of the Workshop on specifying and implementing conversation*, (1999).
- [20] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language," *Software Agents, The AAAI Press* pp. 291-316, (1997).
- [21] D. Steinerark, "FIPA 97 Specification Version 2, Part 2: Agent Communications," *Foundation of Intelligent Physical Agents*, <http://www.fipa.org/spec/FIPA97.html>, (1997).
- [22] ITU-T Rec. M .3010 Principles for a Telecommunications Management Network (TMN), *Study Group IV*, (1996).
- [23] TINA Consortium, Overall Concepts and Principles of TINA, *Document label TB_MDC.018 1.0_94, TINA-C*, (1995).
- [24] David Griffin, George Pavlou and Panos Georgatsos, "Providing Customisable Network Management Services Through Mobile Agentes," <http://www.slpl.lib.mo.us/cco/minutes/103098.htm>, (1998).
- [25] Object Management Group, The Common Object Request Broker: Architecture and Specification (CORBA), Versión 2.0, (1995).
- [26] Mobile Agent System Interoperability Facilities Specification OMG TC Document orbos/97-10-05, <ftp://ftp.omg.org/pub/docs/orbos/97-10-05.pdf>, (1997).
- [27] Somers, Fergal, "Hybrid:unifying centralised and distributed network management using intelligent agents," *IEEE Network Operations and Management Symposium NOMS'96*, pp. 34-43 vol. 1, (1996).
- [28] Jim Hardwicke and Rob Davison, "Software Agents for ATM Performance Management," *Networks Operations and Management Simposium NOMS 98, IEEE*, Volume 2, pp. 313-321, (1998).
- [29] Somers, F. "HYBRID: Intelligents Agents for Distributed ATM Network Management," *IATA Workshop at ECAI'96*, (1996).
- [30] Frei, C. and Faltings, B. "Intelligent Agents for Network Management," *AI for Network Management Systems, IEE Digest No. 97/094*, (1997).
- [31] Gäiti, D. And Boukhatem, N. "Cooperative Congestion Control Schemes in ATM Networks," *IEEE Communications Magazine*, Vol. 34 No. 1, (1996).
- [32] MASIF-RTF Results. Object Management Group, (1998).
- [33] Pavón, J., Tomás, J. Bardout, Y., Hauw, L. H. "CORBA for Network and Service Management in the TINA Framework," *IEEE Communications Magazine*, Vol.36, No 3, (1996).
- [34] W. H. Andrag, and C. W. Omlin, "Distributed Intelligent Multi-Agents for Telecommunication Network Management," http://www.cs.sun.ac.za/projects/tech_reports/
- [35] Simon Steward and Steve Appleby, "Mobile software Agents for Control of Distributed Systems Based on Principles of Social Insect Behaviour," *Proceedings Conference Singapore ICCS'94* pp. 549-553, (1994).
- [36] Gwan Joong Kim, Young Sun Kim and Hyeong Ho Lee, "A design of Management System for ATM Switches Using Mobile Agent Concept," *Global Telecommunications Conference, GLOBECOM 1998. The Bridge to Global Integration. IEEE*, pp. 1694-1698 vol.3. (1998).

- [37] J. Gregoire, "Models and Support Mechanisms for Distributed Management," *Proceedings of the 4th International Symposium on Integrated Network Management*, (1995).
- [38] K. Meyer, M. Erlinger, J. Betser, C. Sunshine, G. Goldszmidt, and Y. Yemini, "Decentralizing Control and Intelligence in Network Management," *Proceedings of the 4th International Symposium on Integrated Network Management*, (1995).
- [39] M. Suzuki, Y. Kiriha, S. Nakai, "Dynamic Script Binding for Delegation Agent," *Proceedings of the 7th IFIP/IEEE International Workshop on Distributed Systems Operation & Management*, (1996).
- [40] T. Magedanz, T. Eckardt, "Mobile Software Agents: A new Paradigm for Telecommunications Management," *Proceedings of IEEE/IFIP 1996 Network Operations and Management Symposium*, (1996).
- [41] Y. Yemini, G. Goldszmidt and S. Yemini, "Network Management by Delegation," *Proceedings of the 2nd International Symposium on Integrated Network Management*, (1991).
- [42] G. Cugola, C. Ghezzi, G. P. Picco, and G. Vigna, "A Characterization of Mobility and State Distribution in Mobile Code Languages," *Proceedings of the 2nd Workshop on Mobile Objects Systems*, (1996).
- [43] Jiann-Liang Chen, Shingfeng Lin, Ming-Yi Lee, "Distributed fault management over ATM networks," *Proceedings of the Sixth IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems*, pp.196-201, (1997).
- [44] Tennenhouse, D. Smith, J., Sincoskie, W.D. and Wetherall, D. "A Survey of Active Network Research," *IEEE Communications Magazine*, pp. 80-85, (1997).
- [45] David L. Tennenhouse, and David Wetherall, "Towards an Active Network Architecture," *Multimedia Computing and Networking*, (1996).
- [46] Dan S. Decasper, Bernhard Plattner, Guru M. Parulkar, Sumi Choi, John D. DeHart, and Tilman Wolf, "A scalable high performance Active Network Node," *IEEE Network*, pp. 8-19, (1999).
- [47] D. Alexander, "Active bridging," *Proc. SIGCOMM'97*, (1997).
- [48] S. Bhattacharjee, "An architecture for active networking," *Proc. INFOCOM'97*, (1997).
- [49] D. Wetherall and D. Tennenhouse, "The ACTIVE IP Options," *Proc. ACM SIGOPS*, (1996).
- [50] David Wetherall, John Guttag and David Tennenhouse, "ANTS: A Toolkit for building and dynamically deploying network protocols," *IEEE OPENARCH'98*, (1998).
- [51] Ulana Legedza, David Wetherall and John Guttag, "Improving the performance of distributed applications using Active Networks," *INFOCOM'98*, pp. 590-599, (1998).
- [52] D.A. Halls and S. G. Rooney, "Controlling the Tempest: Adaptive Management in Advanced ATM Control Architecture," *IEEE JSAC*, Vol. 16, N° 3, pp. 414-423, (1998).
- [53] G. Pujolle and D. Gaïti, "ATM Flow Control Schemes Through a Multi-Agent System," *International Conference on Information Engineering'93, Proceedings of IEEE*, Volume 1, pp. 455-459, (1993).
- [54] Jiann-Liang Chen, Ying-Ping Yu, and S. Lin, "Exploiting Multi-Agent Scheme for Traffic Management in ATM Networks," *Intelligent Control (ISIC), Proceedings, IEEE*, pp. 594-599, (1998).
- [55] German Goldszmidt, and Yechiam Yemini, "Delegated Agents for Network Management," *IEEE Communications Magazine*, Volume 36, 3, pp. 66-70, March (1998).
- [56] G. Goldszmidt, and Y. Yemini, "Distributed Management by Delegation," *IEEE 15th International Conference on Distributed Computing*, pp. 333-340, (1995).
- [57] F. Nait-Abdesselam, N. Agoulmine, and A. Kasiolas, "Agents Based Approach for QoS Adaptation In Distributed Multimedia Applications over ATM Networks," *IEEE International Conference on ATM, ACATM-98*, pp. 319-326, (1998).
- [58] José Luis González-Sánchez and Jordi Domingo-Pascual, "TAP: Architecture for Trusted Transfers in ATM Networks with Active Switches," *ATM'2000 IEEE Conference on High Performance Switching and Routing Joint IEEE ATM Workshop 2000 and 3rd International Conference on ATM (ICATM'2000)*, pp. 105-112 (2000).
- [59] José Luis González-Sánchez and Jordi Domingo-Pascual, "Trusted and Active Protocol over a Distributed Architecture in ATM Networks with agents," *IEEE International Conference on Computer Communication and Networks (IEEE IC3N'2000)*, pp. 484-490 (2000).

Direcciones Universal Resource Locator (URL).

- [URL1] Craig Thompson C. and Odell J., Agent Technology Glossary "OMG Agent WG Technology Green Paper", *Object Management Group*: <http://www.objs.com/agility/tech-reports/9909-agent-glossary.html>.
- [URL2] Grasshopper: The Agent Platform (IKV++): <http://www.ikv.de/products/grasshopper/>
- [URL3] TINA-C <http://www.tinac.com>

- [URL4] (Ramón M. Gómez Labrador, Universidad de Sevilla), <http://www.fie.us.es/~ramon/>
- [URL5] KQML (AgentWeb, Universidad de Maryland en Baltimore): <http://www.cs.umbc.edu/kqml/>
- [URL6] JATLite (CDR, Universidad de Stanford). http://java.stanford.edu/java_agent/html/
- [URL7] FIPA Specifications: <http://www.fipa.com>
- [URL8] France Télécom: <http://www.francetelecom.fr/>
- [URL9] Cluster CLIMATE ACTS <http://www.fokus.gmd.de/research/cc/ima/acts-d5-ac/>
- [URL10]Odyssey y Telescript (General Magic): <http://www.genmagic.com/technology/odyssey.html>
- [URL11]Java Technology Home Page (Sun Microsystems): <http://java.sun.com/>
- [URL12]The Tcl Company (Scriptics): <http://www.scriptics.com/>
- [URL13]Proyecto Tele-MACS <http://www.agentcom.org/agentcom/>
- [URL14]Agentes Inteligentes <http://bogart.sip.ucm.es/~juan/agentes/am.pdf>
- [URL15]Estándares OMG <http://www.omg.org/>
- [URL16]Active Networks home page of MIT <http://www.tns.lcs.mit.edu/activeware/>
- [URL17]Proyecto ACTS IMPACT <http://www.acts-impact.org/impact/>
- [URL18]The Agent Society: <http://www.agent.org/>
- [URL19]The Aglets Software Development Kit (I.B.M. Corp.): <http://www.trl.ibm.co.jp/aglets/>
- [URL20]Mobile Objects and Agents (Open Group): <http://www.caab.opengroup.org/RI/java/moa/index.htm>
- [URL21]Telemedia, Networks and Systems Group <http://www.tns.lcs.mit.edu/>
- [URL22]Agentes y Sistemas Multiagente <http://Montealegre.ei.uvigo.es/Agentes/>
- [URL23]http://www.cetus-links.org/oo_mobile_agents.html