



UNIVERSIDAD DE OVIEDO

Departamento de Informática

Tesis doctoral

**DISEÑO Y EVALUACIÓN DE UNA
PLATAFORMA AUTONÓMICA PARA EL
DESARROLLO DE ACTIVIDADES DE
E-LEARNING SÍNCRONO**

Pelayo Nuño Huergo

Gijón, octubre de 2013



UNIVERSIDAD DE OVIEDO

Departamento de Informática

Tesis doctoral

**DISEÑO Y EVALUACIÓN DE UNA
PLATAFORMA AUTONÓMICA PARA EL
DESARROLLO DE ACTIVIDADES DE
E-LEARNING SÍNCRONO**

presentada por

Pelayo Nuño Huergo

para la obtención del grado de

Doctor por la Universidad de Oviedo

dirigida por **Francisco José Suárez Alonso**
y **Juan Carlos Granda Candás**

Gijón, octubre de 2013

A mi añorado abuelo.

Agradecimientos

Me gustaría empezar agradeciendo el esfuerzo y dedicación empleados por mis directores de tesis, Francisco José Suárez y Juan Carlos Granda, que han sabido orientarme, conducirme, y ayudarme en todo momento. Extiendo este agradecimiento a los miembros del área de Arquitectura y Tecnología de Computadores, en especial a su Señoría Daniel García Martínez por la adjudicación de la beca que me permitió iniciar mi periodo pre-doctoral. Quisiera mostrar mi gratitud en particular a los profesores de la asignatura de Fundamentos de Computadores y Redes por su ayuda y paciencia durante mi primera experiencia docente. Muchas gracias. Finalmente, un recuerdo afectuoso para mi compañero y amigo Francisco, que además de ayudarme, aguantarme y soportarme, fue el que me impulsó a intentar dedicarme a la docencia.

Quiero agradecer también a mi familia su apoyo y comprensión durante este tiempo. En especial a mi madre, que ha aguantado estoicamente mis subidas y bajadas, cambios de humor, momentos de agobio, etc. Muchas gracias. Gracias también a vosotros, Alejandro y David. Por supuesto, mando un sentido recuerdo a los familiares que me vieron iniciar este camino, pero que ya no están aquí para celebrarlo conmigo: Araceli, Pano, Manolo... y, sobretodo, a mi queridísimo y todavía más recordado abuelo Chenchó. Todo mi trabajo te lo dedico a ti.

Retomando el hilo de la docencia, me gustaría recordar igualmente a todos los profesores que, durante mi etapa lectiva, han conseguido mejorar y desarrollar mis aptitudes y estimulado mi interés en el aprendizaje debido a su dedicación, esfuerzo y accesibilidad. En concreto, muchas gracias al Sr. Don Jorge Díez Peláez y, de manera especial, al Sr. Don Juan Carlos Granda Candás, los cuales en diferentes etapas de mi formación han sido piezas fundamentales, fuentes inagotables y magistrales de aprendizaje, y modelos ejemplares de lo que debe ser un profesional docente. Muchas gracias. Incluyo en este recordatorio a los compañeros de estudios que me han servido como referente y guía durante este tiempo, y que igualmente han fomentado mi crecimiento intelectual.

Por último, pero de manera muy afectuosa, quiero agradecer a mis amigos su apoyo y sus ánimos durante estos años. Sin vuestra presencia, física en muchos casos y virtual en algún otro, esta labor no hubiese sido posible. ¡Muchas gracias!

Y como de bien nacidos es ser agradecidos, no puedo olvidar dar las gracias al plan de promoción de la investigación de la Universidad de Oviedo, así como al Banco Santander, por la financiación recibida para desarrollar la presente tesis doctoral.

Resumen

El e-learning síncrono engloba los procesos de enseñanza y aprendizaje a distancia basados en el uso de tecnologías multimedia con interacción en tiempo real. Por tanto, es necesaria una coincidencia temporal entre instructor y alumnos. Este tipo de actividades están cada vez más presentes dentro del ámbito académico y corporativo.

Tradicionalmente, para el desarrollo de las actividades de e-learning síncrono se hacía uso de plataformas de distribución multimedia similares a las utilizadas para la realización de teleconferencias. Hoy en día se converge hacia plataformas que soportan soluciones multimedia que combinan características de las herramientas de teleconferencia y reuniones online, así como del software colaborativo y de gestión personal. Aunque el uso eficiente de los recursos e infraestructuras es una característica presente en la mayoría de las plataformas, aspectos relacionados con la complejidad en el mantenimiento y administración de tales plataformas suponen todavía un reto a superar.

La presente tesis doctoral tiene como objetivo principal diseñar una plataforma de distribución multimedia para el desarrollo de manera eficiente de actividades de e-learning síncrono, y cuya administración, configuración, despliegue y mantenimiento se realicen con la menor intervención humana posible. A tal efecto, la investigación se ha centrado en la aplicación del paradigma de la Computación Autónoma en el ámbito de la distribución de contenidos multimedia en tiempo real interactivos. El germen de la computación autónoma ha sido el hilo conductor de la tesis doctoral puesto que se basa en la idea de auto-gestión como solución al continuo aumento de la complejidad en los sistemas informáticos.

En primer lugar, se han estudiado en profundidad las propiedades inherentes a los sistemas autónomos tratando de identificar aquellas que pueden estar presentes en un sistema de comunicación multimedia. Posteriormente se han destacado las propiedades autónomas cuya implementación en una plataforma de distribución multimedia permitiese facilitar su gestión, así como optimizar su funcionamiento. A raíz del estudio realizado se ha propuesto una clasificación de las propiedades autónomas y sus relaciones, así como una taxonomía para diferenciar las técnicas que permiten implementarlas.

En segundo lugar, se ha diseñado e implementado una plataforma de distribución multimedia que reúne las propiedades autonómicas que se consideran fundamentales de acuerdo al análisis anterior. En concreto, la plataforma implementa técnicas de auto-gestión mediante el uso de protocolos estándares que facilitan su portabilidad, interoperabilidad y extensibilidad.

Finalmente, la plataforma de distribución ha sido utilizada durante el desarrollo de actividades de capacitación en el ámbito de la corporación ArcelorMittal, pudiéndose determinar su impacto en la calidad de la experiencia percibida por los usuarios. Además, se ha desarrollado un modelo de la plataforma que permite estudiar mediante simulación el impacto de nuevas propiedades autonómicas, así como el nivel de escalabilidad y fiabilidad de la plataforma.

Abstract

Synchronous e-learning encompasses a variety of teaching processes and distance-learning techniques which are based on the use of real-time interactive multimedia technologies. Naturally, a temporal coincidence between instructor and students in synchronous e-learning activities is essential. This method of learning has arisen recently, and its use is still growing in both academic and corporate environments.

Typically, synchronous e-learning activities have been carried out through multimedia distribution platforms similar to those used for teleconferencing. Nowadays, there is a convergence towards the development of platforms for multimedia solutions which combine both 1) the features of platforms aimed at the development of teleconferencing and online meetings, along with 2) the features of collaborative software and personal organizers. In general, the majority of the platforms created for the development of conferencing activities make efficient use of network resources and infrastructures, however, several issues related to the complexity of managing and configuring such platforms are still open challenges.

The principal objective of the work carried out for this Ph.D. thesis is the design of an efficient multimedia distribution platform for developing synchronous e-learning activities as well as minimizing the required human intervention to perform management, configuration and deployment tasks. With that purpose in mind, the research has been focused on the Autonomic Computing paradigm, and its application to the field of real-time interactive multimedia data distribution. The central idea of autonomic computing is the promotion of self-management as a solution to the increasing complexity of computer systems, and we have conducted the research guided by this principle.

Firstly, we have analysed the paradigm of Autonomic Computing in order to identify the autonomic characteristics that can be implemented in a multimedia distribution platform. Consequently, we have selected those autonomic characteristics which, when implemented, facilitate a reduction in the platform's management complexity, while also optimizing its performance. The contribution of this research has been twofold. We have proposed a list of autonomic properties and the relations between them. Furthermore, we have presented a taxonomy to classify the wide variety of techniques allowing for implementing autonomic properties.

Secondly, we have designed and implemented a multimedia distribution platform that includes the autonomic properties that we consider mandatory, based on the analysis previously mentioned. Concretely, the platform implements techniques for performing self-management tasks, taking advantage of using standard protocols to ensure that the portability, extensibility and interoperability of the platform are maximised.

Finally, the distribution platform has been deployed to support the training of human resources within a real corporate scenario. This has enabled the impact of the platform on the quality of the experience observed by users to be measured. Furthermore, a simulated model of the platform has been developed which permits measurement of the impact of incorporating new autonomic characteristics into the platform, in addition to measurement of its scalability and resilience levels.

Índice general

Índice de figuras	xI
Índice de tablas	xIII
1. Introducción	1
2. Motivación y objetivos	3
3. Antecedentes	5
3.1. Sistemas autónomos	5
3.1.1. Definición de sistema autónomo	7
3.1.2. Propiedades de los sistemas autónomos	11
3.1.2.1. Auto-configuración (<i>self-configuration</i>)	11
3.1.2.2. Auto-optimización (<i>self-optimization</i>)	12
3.1.2.3. Auto-curación (<i>self-healing</i>)	12
3.1.2.4. Auto-protección (<i>self-protection</i>)	13
3.1.3. Un largo camino por recorrer	13
3.2. Comunicaciones entre grupos dispersos	14
3.2.1. Técnicas multicast a nivel de aplicación	15
3.2.2. Reflectores unicast/multicast	17
3.2.3. Discusión	18
4. Discusión de resultados	21
4.1. Investigación sobre sistemas autónomos	21
4.1.1. Identificación de subpropiedades autónomas aplicables a los SCM	22
4.1.2. Implementación de subpropiedades autónomas en los SCM	26
4.1.3. Taxonomía para clasificar las implementaciones de las subpropiedades autónomas	28
4.2. Diseño e implementación de la plataforma de distribución	30
4.2.1. Diseño de la overlay de la plataforma de distribución	30
4.2.1.1. Red virtual de reflectores	31
4.2.1.2. Red virtual de señalización	32
4.2.1.3. Red virtual de control	32

4.2.2.	Propiedades autonómicas de la plataforma	33
4.2.3.	Resultados de la plataforma	39
4.3.	Investigación sobre el e-learning síncrono	47
4.3.1.	Resultados del uso de la plataforma autonómica y la herramienta e-pSyLon	48
4.3.2.	Resultados sobre nuevos avances y tendencias en e-learning síncrono	50
5.	Conclusiones y trabajo futuro	53
6.	Compendio de publicaciones	57
6.1.	Artículos aceptados	57
6.1.1.	Self-* in Multimedia Communication Overlays	57
6.1.2.	Automatic Deployment of a Communication Mesh for Synchronous e-Learning Activities	58
6.1.3.	Autonomic Platform for Synchronous e-Training in Dispersed Organizations	59
6.1.4.	Security Issues in a Synchronous e-Training Platform	60
6.1.5.	Towards Resilient Synchronous e-Training Platforms	61
6.1.6.	E-pSyLon: A Synchronous e-Learning Platform for Staff Training in Large Corporations	62
6.1.7.	Synchronous E-Learning Tools	63
6.2.	Artículos en proceso de revisión	64
6.2.1.	Resilient Overlay Network for Real-Time Interactive Multimedia Communications in Corporate Networks	64
7.	Informe sobre las publicaciones	65
	Bibliografía	69

Índice de figuras

3.1. Propiedades fundamentales de los sistemas autónomos.	11
3.2. Overlays basada en técnicas multicast a nivel de aplicación. . . .	15
3.3. Overlay compuesta por reflectores.	17
3.4. Comparativa entre las ALM y los reflectores.	18
4.1. Subpropiedades autónomas y su relación con las propiedades fundamentales.	23
4.2. Porcentaje de implementación de cada subpropiedad autónoma.	26
4.3. Taxonomía para clasificar las técnicas self-* en los SCM.	29
4.4. Overlay de la plataforma de distribución de la herramienta e-pSyLon.	31
4.5. Primera fase de la técnica de auto-despliegue: registro de los re- flectores.	34
4.6. Segunda fase de la técnica de auto-despliegue: identificación de la sede corporativa.	35
4.7. Tercera fase de la técnica de auto-despliegue: registro de los parti- cipantes.	35
4.8. Inclusión de un reflector en la red virtual de reflectores.	36
4.9. Exclusión de un reflector de la red virtual de reflectores.	37
4.10. Técnica de auto-curación.	38
4.11. Algoritmo de recuperación del control de la actividad.	39
4.12. Promedio del ancho de banda consumido por cada tipo de tráfico en los reflectores.	40
4.13. Tiempo empleado por la auto-curación para recuperar los parti- cipantes de una sede.	41
4.14. Paquetes de audio perdidos mientras se recuperan los participan- tes de una sede.	42
4.15. Evolución dinámica de la latencia	43
4.16. Estados de la red virtual de reflectores: (a) inicial; (b) tras evento d1; (c) tras evento d2.	44
4.17. Número máximo de participantes estabilizados tras caídas conse- cutivas de reflectores.	45
4.18. Número máximo de participantes e islas multicast estabilizados cuando los recursos de red no están balanceados.	46

4.19. Tiempo empleado por el RP para recuperar el control de la actividad.	47
4.20. Localizaciones geográficas y ancho de banda disponible en el caso de uso de real.	48

Índice de tablas

4.1. Subpropiedades autonómicas en los SCM.	27
4.2. Estado de implementación de cada subpropiedad.	33
4.3. Parámetros de la simulación a nivel de participante.	40
4.4. Eventos simulados para analizar la evolución de la latencia.	44
4.5. Evaluación de la calidad de la experiencia por parte de los participantes.	49

Capítulo 1

Introducción

La investigación en torno a la transmisión de información multimedia a través de Internet ha experimentado un notable crecimiento a lo largo de las últimas décadas. Como resultado han surgido nuevos servicios basados en la comunicación en tiempo real. Un ejemplo de este tipo de servicios es el e-learning síncrono [115], que puede definirse como un proceso de enseñanza y aprendizaje en directo, en tiempo real, interactivo y habilitado por la utilización de medios electrónicos [60]. Las actividades de e-learning síncrono se realizan normalmente entre grupos de participantes geográficamente dispersos. Durante su realización se combinan diferentes medios de transmisión como el audio y vídeo del instructor y los alumnos, contenidos didácticos compartidos, anotaciones de los participantes, mensajería instantánea, etc.; que permiten trasladar las interacciones cara a cara del modelo tradicional de enseñanza a un entorno virtual de educación en línea. Una de sus modalidades más extendidas es el e-training, que consiste en el entrenamiento o capacitación del personal de una empresa dentro del ámbito corporativo.

El requisito principal para el desarrollo de una actividad de e-learning síncrono, desde el punto de vista tecnológico, es que la información multimedia generada por los participantes debe ser retransmitida a todos aquellos que forman parte de la actividad con el menor retraso posible en las comunicaciones, utilizando el mínimo ancho de banda, y sin incrementar en exceso el consumo de recursos computacionales en los equipos de los participantes. Para satisfacer las condiciones anteriores, las actividades de e-learning síncrono se desarrollan sobre complejas plataformas de distribución multimedia, formadas por múltiples elementos distribuidos y con severas restricciones temporales, que en algunos casos pueden requerir de personal especializado para su configuración y administración, lo que repercute en un sobrecoste para la entidad responsable de la actividad. Cuando la configuración o la administración de este tipo de plataformas no resulta transparente y requiere de intervención humana se convierte en un proceso tedioso y propenso a que se cometan errores que pueden dificultar y retrasar la realización de las actividades programadas.

El aumento en la complejidad de los sistemas informáticos es uno de los problemas que más se han debatido en la comunidad científica en los últimos años. A este respecto, Paul Horn, vicepresidente de IBM, y director de su departamento de investigación, presentó en 2001 el paradigma de la **Computación Autónoma** (CA). El objetivo de su manifiesto era alertar a la comunidad científica y a la industria de las Tecnologías de la Información (TI) acerca de una inminente crisis del software debida a la creciente complejidad de los sistemas de información [55]. La esencia de la computación autónoma es la auto-gestión (*self-management*), mediante la que es posible liberar a los administradores de sistemas de los detalles sobre su funcionamiento y mantenimiento, así como proporcionar a los clientes un sistema fiable que se encuentre operativo siempre a máximo rendimiento. Para ello, un sistema autónomo debe ser capaz de ajustar su comportamiento en función de los cambios que se produzcan en sus componentes, de las demandas de servicio, de las condiciones externas, o en presencia de incidencias hardware o software tanto maliciosas como fortuitas.

Dentro del ámbito de las comunicaciones multimedia el paradigma de la CA ha tenido gran incidencia. Así pues, es posible encontrar numerosos trabajos recientes relacionados con las tecnologías *peer-to-peer* (P2P) que exhiben alguna de las propiedades de los sistemas autónomos: sistemas auto-organizados [22], auto-distribuidos [135], auto-regulados [70] y sistemas tanto auto-diagnosticables como auto-recuperables [22, 70, 135]. En la parcela de las plataformas orientadas al e-learning síncrono existen soluciones que, aunque satisfacen alguna de las propiedades de los sistemas autónomos, no están conceptualmente diseñadas conforme a dicho paradigma, por lo que los comportamientos y funcionalidades que exhiben podrían ser ampliados o complementados [54, 76]. Dichas soluciones presentan puntos débiles y pueden ser mejoradas. Por ejemplo, la mayoría son soluciones propietarias que no se basan en protocolos estándares, impidiendo su extensibilidad hacia otro tipo de aplicaciones y otros dispositivos. Esta tesis doctoral surge con la intención de analizar el paradigma de la CA aplicado a los sistemas de distribución multimedia en tiempo real interactivos para mejorar las soluciones existentes, extrayendo todas aquellas características que, aplicadas sobre dichos sistemas, permitan incrementar su eficiencia de cara a la realización de actividades de e-learning síncrono en el ámbito corporativo y académico.

Se presenta la tesis como un compendio de publicaciones, estando organizada como sigue: En el capítulo 2 se plantean su motivación y objetivos. El capítulo 3 presenta los antecedentes sobre las temáticas que vertebran esta tesis. Los resultados obtenidos y las contribuciones aportadas se describen en el capítulo 4. Las conclusiones y trabajo futuro se comentan en el capítulo 5. Finalmente, el capítulo 6 contiene los artículos que forman el compendio de publicaciones, mientras que el informe sobre su impacto se detalla en el capítulo 7.

Capítulo 2

Motivación y objetivos

La presente tesis doctoral se centra en mejorar las plataformas de distribución de contenidos multimedia generados durante las actividades de e-learning síncrono. Tanto el e-learning como la distribución de contenidos multimedia son líneas de investigación que han crecido notablemente en los últimos años, y de gran interés para el área de Arquitectura y Tecnología de Computadores de la Universidad de Oviedo. En tal contexto se tiene como precedente la tesis del Dr. Granda Candás, la cual dio como fruto la herramienta de e-learning síncrono e-pSyLon, para la realización de actividades de teleenseñanza en el ámbito académico y corporativo [39]. Además, también se desarrolló una plataforma de distribución de contenidos multimedia básica para desarrollar actividades de e-learning síncrono entre grupos dispersos [40]. Sin embargo, una plataforma tan básica adolece de varias características autonómicas que permitirían realizar actividades de e-learning síncrono de manera más eficiente y fiable. La tesis representa un trabajo continuista en ese sentido y, de forma concreta, plantea los siguientes objetivos:

- Identificar, clasificar y caracterizar aquellas propiedades que conforman el paradigma de los sistemas autonómicos cuya presencia sería deseable en los sistemas de comunicaciones en tiempo real interactivos, y en especial, en las plataformas de distribución orientadas al e-learning síncrono.
- Desarrollar una plataforma de distribución para la herramienta de e-learning síncrono e-pSyLon que implemente las propiedades más relevantes de los sistemas autonómicos derivadas del análisis anterior, de tal manera que la experiencia de uso de instructores y alumnos se vea favorecida.
- Evaluar el comportamiento conjunto de la plataforma de distribución y la herramienta de e-learning síncrono e-pSyLon, valorando la incidencia y efectividad de las propiedades implementadas en la calidad de la experiencia de uso de los usuarios finales.
- Aplicar la tecnología desarrollada en un entorno real, bien en procesos académicos o de formación de los recursos humanos de una corporación.

Capítulo 3

Antecedentes

En este capítulo se realiza un breve análisis de los conceptos fundamentales en los que se basa esta tesis doctoral. En primer lugar se describe el paradigma de la computación autónoma. A continuación, se analizan las técnicas que permiten realizar comunicaciones multimedia de manera eficiente entre grupos de participantes dispersos.

3.1. Sistemas autónomos

"A la industria de las tecnologías de la información (TI) le encanta demostrar que lo imposible puede resultar posible.

Eliminamos barreras y establecemos nuevos límites con asombrosa regularidad. Pero ahora nos enfrentamos a un problema que está surgiendo desde la raíz de nuestro éxito, sin que suficientes de nosotros estemos concentrados en resolverlo.

Este que nos ocupa, más que cualquier otro problema referente a las TI, evitará que alcancemos la siguiente era de la informática si permanece sin resolverse. Curiosamente, tiene poco que ver con las barreras que tradicionalmente nos han preocupado.

No tiene que ver con mantener el ritmo de la ley de Moore, sino más bien con afrontar las consecuencias de las décadas en las que la cual ha prevalecido. Tampoco está relacionado con cuántos bits pueden llegar a almacenarse en una pulgada cuadrada, ni con el nivel de precisión con el que es posible operar sobre un chip de silicio. De hecho, la continua obsesión con desarrollar tecnologías más reducidas, rápidas y baratas es en realidad una distracción.

El obstáculo es la complejidad. Tratar con ella es el mayor desafío al que se enfrenta la industria de las tecnologías de la información.

Es nuestro próximo gran reto."

Paul Horn, director de investigación y vicepresidente de IBM, Marzo de 2001.

El 8 de Marzo de 2001, Paul Horn, vicepresidente senior de IBM y director de su departamento de investigación, comenzó con la declaración anterior la presentación del paradigma de la Computación Autónoma (CA) a la Academia Nacional de Ingeniería en la Universidad de Harvard. El objetivo de su manifiesto era alertar a la comunidad científica y a la industria de las TI acerca de una inminente crisis del software cuyo origen era el continuo aumento de la complejidad en los sistemas de información [55]. A medida que se ha mejorado el rendimiento de los componentes hardware los sistemas han visto aumentada su complejidad. Los desarrolladores de software pueden hacer uso de un poder computacional cada vez mayor que da lugar a aplicaciones y entornos más complejos y sofisticados. De igual manera, la administración de sistemas a lo largo de la última década se ha convertido también en una tarea cada vez más difícil. La complejidad de estos nuevos sistemas, creciente de forma continua, parece aproximarse al umbral de la capacidad operativa del ser humano, ya que cada vez se necesitan profesionales más especializados, y con más experiencia, para instalar, configurar, operar o mantener dichos sistemas.

Para hacer más acuciente la crisis que se aproxima, la disponibilidad de los profesionales capacitados es crítica ya que, si la complejidad de los sistemas prosigue en su crecimiento y expansión, no habrá suficientes profesionales para cubrir la demanda reclamada por la industria. Esto es debido a que, al tiempo necesario para la formación de un profesional capacitado hay que añadirle la cantidad de modelos de negocio cuya infraestructura debe estar disponible a tiempo completo, 24 horas al día, 7 días a la semana, lo que desembocaría en el colapso vaticinado por Horn. Por otra parte, aunque la cobertura de profesionales fuese suficiente para satisfacer la demanda industrial, se estaría muy lejos de una solución al problema de partida. En algunos casos la administración de sistemas se ha vuelto demasiado compleja y, por tanto, propensa a errores. Los profesionales están sometidos a una presión cada vez mayor que no está acorde con un incremento del tiempo disponible para la toma de decisiones, lo que aumenta la probabilidad de que se cometan errores que desemboquen en cortes en el servicio que originen serias consecuencias económicas a las empresas, y pérdida de fidelidad por parte de sus clientes.

Adicionalmente, el análisis del coste originado a las empresas (factor común decisivo en cualquier tipo de modelo industrial) ejerce como indicador claro de la necesidad de trasladar el modelo actual de desarrollo de sistemas hacia el paradigma de la CA. Cada vez las empresas dan más relevancia al baremo denominado retorno de la inversión (ROI), que consiste en la relación entre el beneficio obtenido por una empresa y la inversión realizada para tal efecto. La complejidad de los sistemas, y la dependencia de profesionales cada vez más cualificados da lugar a un incremento del coste total de propiedad del sistema (TCO), que repercute directamente en el aplazamiento del retorno de la inversión inicial [109]. Además, a los costes de propiedad y adquisición del sistema, hay

que sumarle los derivados de las labores de prueba y mantenimiento, que han crecido igualmente en complejidad ocupando una parcela de tiempo en el ciclo de actividad de las empresas cada vez mayor.

Horn y otros autores coinciden en señalar un cambio en el paradigma del diseño de sistemas como único mecanismo para revertir la situación y superar la crisis que se aproxima:

"Es hora de diseñar y construir sistemas capaces de ejecutarse ellos mismos, ajustarse a circunstancias variables, y administrar sus recursos para manejar de manera más eficiente las cargas de trabajo que les asignamos. Estos sistemas autónomos deben anticipar las necesidades de los usuarios y permitirles concentrarse en qué quieren conseguir, en lugar de plantearse cómo manipular el sistema informático para conseguirlo [55]".

"Solo existe una solución: la tecnología debe gestionarse ella misma. La tecnología debe dejar de requerir la presencia de un ser humano que haga todo por ella, y pasar a comportarse de una manera más inteligente, tal y como esperábamos que fuera, empezando a hacer frente a sus propias necesidades [127]".

3.1.1. Definición de sistema autónomo

La esencia de los sistemas de computación autónoma es la idea de auto-gestión. Mediante ella es posible, por un lado, liberar a los administradores de los detalles de funcionamiento y mantenimiento del sistema, y por otro, proporcionar a sus clientes un sistema fiable que se encuentre operativo a máximo rendimiento en todo momento (modelo 24x7). Los sistemas autónomos ajustan su comportamiento en función de los cambios que se producen en sus componentes, de las cargas de trabajo, de las demandas de servicio, de ciertas condiciones externas como la saturación de las líneas de comunicación, o en presencia de fallos hardware o software tanto malintencionados como involuntarios. Los sistemas autónomos, por tanto, deben realizar por sí mismos tareas de monitorización, actualización de componentes, reconfiguración, regresión y detección y aislamiento de errores, entre otras [65].

Un enfoque que puede servir para comprender el propósito de la CA es profundizar en el origen de su denominación. Cuando se presentó este concepto se eligió deliberadamente un término con connotaciones biológicas. El paradigma de la CA se inspira en el sistema nervioso autónomo (SNA), uno de los ejemplos más sofisticados de capacidad de auto-gestión en la naturaleza. El SNA es un componente del sistema nervioso constituido por un conjunto de neuronas y nervios que controlan funciones de diferentes sistemas del organismo. Su

propósito consiste en mantener la situación de homeostasis ¹ del organismo y efectuar las respuestas de adaptación ante cambios del entorno. Como su propio nombre indica, el SNA no se encuentra sujeto al control voluntario o consciente [82]. El SNA regula la respiración, la circulación sanguínea, la digestión, el metabolismo, la secreción glandular, la temperatura corporal, la reproducción celular y, además, coordina todas estas funciones vitales para mantener la homeostasis.

En el SNA se integran dos subsistemas principales: el sistema simpático y el parasimpático. La actividad simpática del SNA está dirigida a situar al individuo en una situación de defensa ante circunstancias de peligro, real o potencial. La estimulación simpática conduce a variaciones de las funciones internas, destinadas a proteger la integridad del organismo como un todo y a garantizar la supervivencia, que se definen como respuesta de alarma. Algunos de los fenómenos corporales más evidentes de esta respuesta son: dilatación pupilar para aumentar el campo visual, piloerección para simular un mayor tamaño corporal, aumento de la actividad cardíaca y de la presión arterial para proporcionar un mayor flujo sanguíneo en los músculos, o broncodilatación para aumentar la entrada de aire a los pulmones. Por el contrario, la actividad parasimpática del SNA está relacionada con funciones protectoras y de conservación, que favorecen el correcto funcionamiento de los diferentes sistemas internos. Por ejemplo: constricción pupilar para proteger la retina de un exceso de iluminación, disminución de la frecuencia cardíaca para evitar una actividad excesiva, o broncoconstricción para proteger los pulmones.

Tal y como se ha enunciado, la característica principal del SNA es controlar y regular la homeostasis. Un sistema homeostático, por ejemplo una gran corporación o empresa industrial, es un sistema abierto que mantiene su estructura y funcionalidades mediante múltiples ajustes dinámicos que son rigurosamente controlados por mecanismos de control independientes. Se trata de un sistema que reacciona ante cada cambio en su entorno, o perturbación aleatoria, a través de una serie de respuestas, equivalentes y compensadas, en dirección opuesta respecto a las variaciones que provocaron la alteración. El propósito de dichas respuestas es mantener el equilibrio del sistema. Trasladando el modelo del SNA hacia el paradigma de la CA, los sistemas de computación disponen de mecanismos para que cualquier cambio en sus condiciones esenciales de operación implique respuestas automáticas en su comportamiento, o régimen de funcionamiento, para que dicho sistema mantenga un estado de equilibrio respecto a su entorno operacional. Las variaciones de su entorno operacional pueden ser tanto externas (ataques de agentes ajenos al sistema) como internas (excesivo consumo de memoria o de CPU). Asegurar un estado de equilibrio

¹Característica de los seres vivos por la cual, mediante la acción del metabolismo, pueden regular las funciones que existen dentro de ellos para mantener una condición estable y constante [18].

respecto a su entorno operacional implica la supervivencia del sistema ya que es capaz de protegerse a sí mismo, recuperarse ante fallos, reconfigurarse según los cambios que se produzcan en su entorno, y todo ello manteniendo un rendimiento operacional cercano a su régimen de funcionamiento óptimo [89].

Los sistemas autónomos desde su origen conceptual están destinados a funcionar mediante una serie de directrices de alto nivel indicadas por los administradores del sistema. Estas directrices reflejan objetivos de la empresa o regímenes mínimos de rendimiento a satisfacer, que el propio sistema posteriormente es capaz de desglosar automáticamente en múltiples decisiones y acciones de más bajo nivel. Así, los usuarios del sistema podrán concentrarse exclusivamente en qué quieren obtener de él, y no tienen que preocuparse respecto a cómo hay que interactuar con el sistema para conseguirlo. Por tanto, la adaptación al sistema se produce de manera inconsciente, sin requerir ningún esfuerzo o modificación del comportamiento del usuario [113].

Tras la definición formal de sistema autónomo existen una serie de propiedades o criterios fundamentales que un sistema debe satisfacer para ser considerado como tal [55]:

1. **Para poder considerarse autónomo, un sistema necesita conocerse a sí mismo.** Todo aquello que un sistema no reconozca como propio no puede ser monitorizado ni controlado, lo que implica una debilidad desde el punto de vista de su autonomía. Es imprescindible que un sistema conozca con exactitud aspectos como los componentes que lo integran, estado actual, capacidades, recursos que puede compartir o debe requerir, relaciones con otros sistemas, etc. Esta propiedad es recursiva, y por tanto extensible a todos los componentes que integran un sistema autónomo.
2. **Conoce su entorno operacional y actúa en consecuencia respecto al contexto en el que se encuentra.** Refina y adapta el proceso de diálogo e interacción con otros sistemas para obtener el grado de interrelación más beneficioso a nivel global. Este proceso de diálogo incluye la negociación de sus recursos disponibles, que pueden ser cedidos a otro sistema que los solicite para así mejorar el rendimiento general del entorno operacional.
3. **Se configura y reconfigura a sí mismo bajo condiciones variables e impredecibles.** Las modificaciones del sistema se realizan de manera automática y dinámica, estando siempre destinadas a adecuarse a cambios que hayan ocurrido en su entorno operacional y con el propósito de alcanzar una mejor adaptación a dicho entorno.
4. **Nunca se asienta en un *statu quo*, siempre busca maneras de optimizar las tareas que desempeña.** El sistema dispone de unos

objetivos predefinidos de rendimiento que compara, en base a métricas propias de evaluación, con el rendimiento obtenido en la práctica, y realiza modificaciones automáticas para ir acercándose al ideal, e incluso superarlo si los objetivos de partida fuesen pesimistas.

5. **Es robusto y se fortalece más aun durante su tiempo de actividad.** En la presencia de eventos habituales o extraordinarios que puedan causar un mal funcionamiento de alguno de sus componentes, o del sistema en general, los sistemas autonómicos descubren y/o se recuperan de dichos eventos de manera transparente, manteniendo su nivel de funcionamiento cercano al óptimo. Para ello determinan la mejor solución en base a la información disponible en el momento de actuar y, en el caso de enfrentarse a incidencias desconocidas, sientan precedentes para futuras actuaciones semejantes, al igual que la memoria inmunológica de los organismos vivos.
6. **Es capaz de protegerse de amenazas externas.** Siguiendo con la analogía entre un sistema de computación autonómico y un organismo vivo, el mundo virtual supone un escenario igual de peligroso y lleno de riesgos que el mundo real. Por tanto, los sistemas autonómicos se protegen frente a amenazas presentes en su entorno. Son capaces de detectar, identificar y contrarrestar ataques contra la seguridad e integridad del sistema manteniéndolo invulnerable.
7. **No tienen cabida dentro de entornos herméticos.** Tan importante es la gestión de sí mismos, como mantener el conocimiento exacto de su entorno operacional en base a interacciones con otros sistemas. Las comunicaciones con otros sistemas suponen el mismo reto que las comunicaciones entre comunidades de individuos de diferentes culturas: heterogeneidad a diferentes niveles. Para poder superar ese obstáculo, se debe hacer uso de algún mecanismo conocido y aceptado por todos los implicados en la comunicación. En el caso de los sistemas autonómicos, deben estar implementados sobre estándares abiertos.
8. **Se anticipa a necesidades futuras, disponiendo siempre de los recursos necesarios.** Son capaces de dictaminar las necesidades reales de los usuarios, anticipando y planificando qué recursos serán necesarios para llevarlas a cabo. Identifican situaciones potencialmente límite que sobrepasan sus capacidades actuales, y se adaptan a las mismas para poder afrontarlas apropiadamente mediante interacciones con otros elementos de su entorno.
9. **Mantiene oculta a sus usuarios la complejidad intrínseca necesaria para satisfacer las propiedades anteriores.** El propósito por el que surge el concepto de la CA era hacer frente a la complejidad creciente

de los sistemas de información, por lo que esta propiedad forma parte de su propia naturaleza. Los sistemas autónomos evitan a los usuarios un proceso de aprendizaje excesivo sobre cómo utilizarlos, interactuar con ellos o interpretar sus resultados.

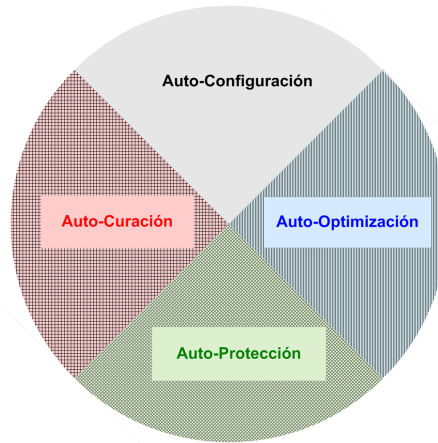


Figura 3.1: Propiedades fundamentales de los sistemas autónomos.

3.1.2. Propiedades de los sistemas autónomos

El propósito principal de los sistemas autónomos es la auto-gestión (self-management). Para cumplir dicho propósito, y satisfacer las características enunciadas en el apartado anterior, los sistemas autónomos se basan en cuatro propiedades principales, conocidas como *self-properties* [62, 65, 89, 109]. En la figura 3.1 se muestran las propiedades fundamentales de los sistemas autónomos.

3.1.2.1. Auto-configuración (*self-configuration*)

Uno de los fundamentos principales de los sistemas autónomos es la auto-configuración, que se refiere a la adaptación automática y dinámica del sistema a su entorno y a los distintos cambios que se produzcan en él. La auto-configuración incluye varios aspectos de la gestión de sistemas. En primer lugar, los sistemas autónomos se configuran y reconfiguran por sí mismos de acuerdo a unas directrices de alto nivel que, como se ha comentado anteriormente, definen los objetivos del sistema. Otro aspecto que cubre la auto-configuración es la adición de componentes, que pueden ser añadidos

dinámicamente sin que implique una detención en su servicio. Finalmente, la integración de los nuevos componentes se realiza de manera autónoma, ya que se incorporan interaccionando con el resto del sistema de tal forma que el conjunto va conformando progresivamente la nueva visión del sistema.

3.1.2.2. Auto-optimización (*self-optimization*)

Los sistemas autónomos controlan y ajustan el uso de sus recursos de manera automática. Continuamente buscan un funcionamiento más eficaz, identificando y actuando en función de circunstancias o eventos que les permitan hacerse más eficientes en términos de rendimiento o coste. Dentro de la auto-optimización hay gran variedad de formas de actuación. La distribución de cargas de trabajo, la optimización de rutas o arquitecturas de red o la reinterpretación de consultas sobre bases de datos son solo algunos ejemplos. En resumen, los sistemas autónomos buscan de manera activa optimizar la ejecución de sus actividades durante el ciclo de vida de las mismas. Para ello determinan, verifican, prueban y aplican, de manera paralela a su prestación de servicio, modificaciones que tengan una incidencia positiva en la prestación de dicho servicio y en su relación con el entorno operacional. La auto-optimización y auto-configuración son parcelas adyacentes, ya que en algunos casos una optimización del sistema puede requerir una reconfiguración total o parcial del mismo. Así mismo, también está relacionada con la auto-protección puesto que mantener el óptimo funcionamiento de un sistema es una forma de protegerlo.

3.1.2.3. Auto-curación (*self-healing*)

La auto-curación se basa en que los propios sistemas identifican, diagnostican y se recuperan de interrupciones y errores, tanto a nivel software como hardware, que comprometan el desarrollo de su actividad. Utilizando un símil con los seres humanos, aprovechan el conocimiento adquirido durante su ciclo de actividad (memoria inmunológica) para identificar síntomas y determinar el modo de afrontarlos o recuperarse de ellos, seleccionando el mecanismo *curativo* que traslade al sistema a un estado lo más cercano posible a su estado de equilibrio. En el caso de que detecten síntomas de los que no tiene ningún conocimiento previo, pueden requerir la ayuda de un administrador humano (inmunización pasiva). La auto-curación utiliza diversas técnicas para su puesta en práctica como son el aislamiento de elementos defectuosos, o bien su reemplazamiento o reparación sobre la marcha. Algunas de las técnicas anteriores podrían requerir de una reconfiguración del sistema, luego auto-curación y auto-configuración son también parcelas adyacentes.

3.1.2.4. Auto-protección (*self-protection*)

La auto-protección tiene dos enfoques, reactivo y proactivo. Desde un enfoque reactivo, los sistemas autonómicos se protegen de problemas interrelacionados a gran escala que pueden surgir a raíz de ataques maliciosos por parte de terceros, o de fallos en cascada que no han podido ser solventados por las medidas de auto-curación, o que implican la actuación combinada de ambas propiedades. Así pues, detectan comportamientos anómalos cuando estos ocurren, tomando decisiones para hacerse a sí mismos menos vulnerables frente a situaciones de riesgo como son accesos o usos no autorizados, ataques de denegación o degradación de servicio, virus, etc. Por otro lado, a través del enfoque proactivo los sistemas autonómicos tratan de anticipar las situaciones de riesgo, bien basándose en la experiencia adquirida, o bien en función de circunstancias concretas, para tomar las medidas necesarias para evitarlas o al menos tratar de mitigar sus consecuencias.

3.1.3. Un largo camino por recorrer

El paradigma de la computación autonómica no consiste simplemente en una nueva parcela a ocupar en el campo de la informática que se pueda cubrir de manera aislada mediante aportaciones individuales e inconexas. Se trata de un desafío muy importante y el reto todavía permanece abierto. El desarrollo de sistemas autonómicos requiere, por tanto, de la cooperación de las grandes empresas y de la comunidad científica para poder convertirse en una realidad mediante una combinación de cambios en los procesos, nuevas tecnologías y arquitecturas, y la plena adecuación de la industria a los estándares [89].

No obstante, el cambio que debe afrontarse para convertir el CA en una realidad **no se trata de una revolución, sino de una evolución** [38], que engloba tanto factores tecnológicos como factores a nivel humano. Los factores tecnológicos hacen referencia a protocolos, estándares y desarrollos que van construyendo paso a paso el paradigma de la CA. A nivel humano, el factor principal es la confianza. Pasar de un modelo de computación supervisado a un modelo autonómico puede causar recelo, especialmente en tareas críticas o de riesgo. A medida que la tecnología evolucione, los usuarios se irán acostumbrando a una cada vez menor toma de decisiones respecto al sistema. La confianza en la tecnología evolucionará al mismo ritmo que la propia tecnología, por lo que el cambio se irá realizando de manera progresiva e inconsciente [79]. En resumen, aunque la computación autonómica se ha convertido en un paradigma cada vez más interesante y popular, todavía permanece como un tema relativamente inmaduro. A medida que más disciplinas adopten este paradigma la investigación crecerá, y se irá recorriendo el camino hasta consolidarlo definitivamente [59].

3.2. Comunicaciones entre grupos dispersos

El propósito de todo sistema de comunicación multimedia (SCM) entre grupos geográficamente dispersos, y en particular de las plataformas orientadas a la comunicación en tiempo real interactiva como es el caso del e-learning síncrono, es ofrecer un servicio de transporte de información estable y con un consumo eficiente de los recursos disponibles. Además, la latencia debe ser mínima con el fin de favorecer una comunicación interactiva. La alternativa óptima para satisfacer tales objetivos en las comunicaciones multimedia entre grupos dispersos es utilizar IP multicast [28]. Este modelo de comunicación representa el mejor escenario de distribución posible, puesto que no existe replicación de datos a nivel de red y asegura que la información alcanza su destino con la menor latencia posible [66].

Aunque existen casos de éxito en el uso de IP multicast para realizar comunicaciones entre grupos geográficamente dispersos, por ejemplo en el ámbito de las conferencias [134], su utilización es muy limitada. Un inconveniente es que los enrutadores deben ser configurados apropiadamente para soportar las comunicaciones multicast. No obstante, el principal problema del uso de IP multicast es que plantea vulnerabilidades que pueden ser explotadas para provocar un consumo masivo de recursos de red [29].

Las cuestiones anteriores han desembocado en que, por lo general, los proveedores de servicio de Internet no habilitan la posibilidad de comunicarse mediante IP multicast en sus redes, por lo que no es posible llevar a la práctica dicho modelo de comunicación para comunicar grupos geográficamente dispersos. No obstante, existen técnicas de comunicación que tratan de emular el funcionamiento de una red basada en IP multicast, aunque la eficiencia en la entrega de los datos se ve penalizada respecto al modelo IP multicast nativo. Estas técnicas se basan en el despliegue de unas redes virtuales, denominadas overlays, que permiten la abstracción de la red física. Una overlay es el mecanismo subyacente que posibilita la realización de comunicaciones multipunto eficientes en los SCM [129]. Por tanto, son el núcleo de toda plataforma de distribución, puesto que gestionan y soportan el intercambio de información.

Las overlays son sistemas de transmisión complejos que se despliegan a nivel de aplicación en todos, o en parte, de los computadores involucrados en la comunicación, posibilitando el intercambio de información en tiempo real entre múltiples miembros. Su régimen de funcionamiento viene impuesto por una serie de restricciones. En primer lugar, deben ser capaces de sortear la heterogeneidad de las infraestructuras de red para poder desplegarse y operar sobre toda clase de redes. En segundo lugar, deben minimizar aspectos de la comunicación como la latencia o la variabilidad de ésta (*jitter*). Por último, deben hacer un uso eficiente de los recursos disponibles maximizando la calidad del servicio.

El principal problema para cumplir apropiadamente con tales restricciones es que habitualmente los SCM buscan un compromiso entre dos objetivos fundamentales que resultan contrapuestos a nivel de la overlay: minimizar el retardo en las comunicaciones y maximizar la escalabilidad [106].

Un amplio catálogo sobre alternativas que permiten mejorar la comunicación entre grupos geográficamente dispersos mediante el uso de overlays se puede encontrar en [32, 129], donde destacan principalmente dos enfoques; las técnicas multicast a nivel de aplicación (*Application Layer Multicast*, ALM) y los reflectores unicast/multicast.

3.2.1. Técnicas multicast a nivel de aplicación

Las técnicas multicast a nivel de aplicación engloban a las plataformas de distribución en las que tanto la entrega de la información como las labores de administración se realizan a nivel de aplicación en los propios miembros de la overlay [8, 92], es decir, en los equipos de los participantes en el caso de una actividad de e-learning.

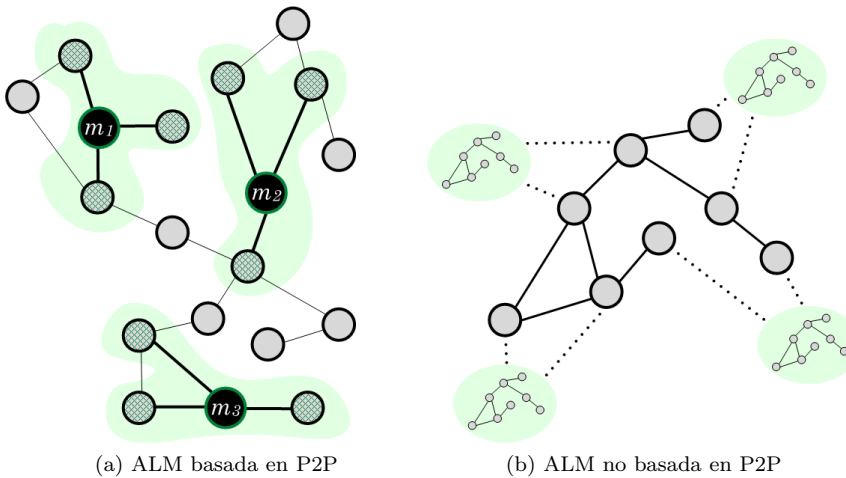


Figura 3.2: Overlays basada en técnicas multicast a nivel de aplicación.

Existen dos modelos principales de ALM como representa la figura 3.2. En primer lugar, las ALM que se despliegan siguiendo el modelo *peer-to-peer* (P2P) [94, 130]. La principal diferencia entre las ALM basadas en P2P y el resto es el grado de conocimiento acerca de la overlay. Cada miembro propaga su visión local de la overlay entre sus vecinos según el modelo P2P, por tanto, la visión de la overlay por cada miembro es parcial, y probablemente incompleta,

tal y como se ilustra en la figura 3.2a con los miembros m_1 , m_2 y m_3 . Esto promueve overlays con relaciones de adyacencia muy volátiles, que son refinadas periódicamente en base a la disponibilidad de la información o la contribución de los miembros en la distribución de la información a través la overlay [118, 135]. Por contra, como muestra la figura 3.2b, en las ALM no basadas en el modelo P2P la visión de la overlay es más estricta y general, ya que todos los miembros tienen un conocimiento completo de la estructura de la overlay para poder decidir la ruta óptima por la que se debe transmitir la información [22].

Un exhaustivo análisis sobre los diferentes tipos de ALM se puede encontrar en [56]. El autor clasifica las ALM en función de varios aspectos como su topología, malla [22] o árbol [63]; la administración de la overlay, centralizada [92] o distribuida [8]; y su propósito, minimizar la longitud de la ruta hacia un destino concreto (*Source Specific Tree*) [69] o minimizar la longitud media de las rutas que componen la overlay (*Shared Tree*) [108].

Desde el punto de vista de la topología de la overlay han surgido ALM híbridas que combinan varios aspectos. Por ejemplo, existen ALM que mezclan las topologías de árboles y mallas dando lugar a overlays organizadas mediante agrupaciones (*clustering*) [114], donde cada una de las hojas del árbol principal de distribución es un conjunto de participantes interconectados a través de una malla. Otras ALM híbridas consisten en múltiples árboles solapados que dan lugar a varias rutas para la distribución de la información [19].

La utilización de ALM presenta varias ventajas. Son muy flexibles, lo que permite extender su despliegue para dar cobertura a actividades con un gran número de participantes. Esta flexibilidad radica en la independencia respecto a entidades intermedias, ya que los propios miembros de la overlay realizan las labores de control y gestión. A su vez, esta independencia da lugar a otra ventaja notable como es su fiabilidad puesto que, sea cual sea la entidad de la overlay que sufra un error, la overlay nunca queda huérfana de miembros capaces de administrarla, ni se ve interrumpida la retransmisión de información multimedia. El principal ámbito de aplicación de las ALM son las plataformas de distribución a gran escala de contenidos en tiempo real (*live streaming*) [70] y de vídeo bajo demanda (VoD) [50].

Sin embargo, las ventajas anteriores conllevan un aumento en la latencia en las comunicaciones. Además, existen también diferencias en la latencia percibida por diferentes participantes dentro de una misma actividad en función de la distancia con el origen de datos. Este hecho puede dificultar notablemente la realización de actividades en tiempo real interactivas.

3.2.2. Reflectores unicast/multicast

La técnica basada en reflectores unicast/multicast engloba el conjunto de plataformas en las que existe un elemento en cada grupo de participantes, denominado reflector o *relay*, que sirve de intermediario en las comunicaciones con los restantes grupos, actuando como una pasarela por la que circula la información de su grupo hacia el exterior y viceversa [53, 116]. Los reflectores se interconectan entre sí mediante una topología *full-mesh* (es decir, todos con todos), dando lugar a una overlay por la que transcurren las comunicaciones entre los distintos grupos. Por tanto, el reflector conoce de antemano un conjunto de destinos a los que debe emitir, así como desde dónde recibir información multimedia. El tráfico proveniente de un miembro particular es retransmitido por el reflector hacia el resto de miembros de su grupo y hacia la red externa y viceversa. Este tipo de plataformas se pueden interpretar como la creación de un conjunto de túneles puntuales entre el reflector y cada uno de los destinos de información, donde los paquetes multicast son encapsulados como paquetes IP y enviados normalmente.

Tanto en las primeras versiones de este tipo de plataformas como en algunas ampliaciones no se implementaban características autonómicas. Por ejemplo, la administración y configuración de la overlay de reflectores requería intervención humana y se carecía de técnicas de auto-curación [54, 117]. Sin embargo, en plataformas más recientes sí se han incorporado características autonómicas como la auto-organización o la auto-curación [128], incluso basadas en protocolos estándares [67, 90] como el *Real-Time Streaming Protocol* (RTSP) [103].

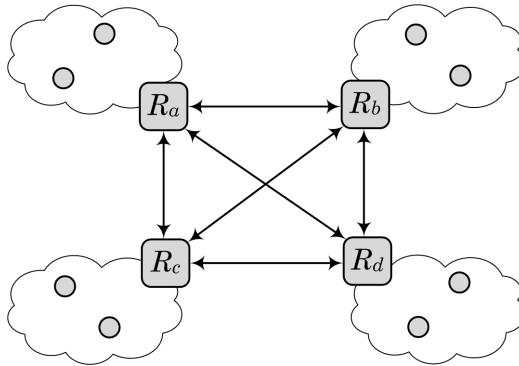


Figura 3.3: Overlay compuesta por reflectores.

El uso de reflectores ofrece diversas ventajas. La principal es que permite limitar a dos el número de elementos por los que circula la información entre grupos cualesquiera, tal y como se muestra en la figura 3.3. Por tanto, al minimizar el número de elementos por los que circula la información se minimiza también la latencia, lo que origina que estas plataformas sean más indicadas que las ALM para soportar el desarrollo de actividades multimedia en tiempo real interactivas. De hecho, existen precedentes del uso de reflectores en el ámbito del e-learning síncrono [76]. Además, se trata de una solución transparente en ámbitos donde esté habilitada la comunicación IP multicast, puesto que el reflector se comporta como un miembro más del grupo multicast. Otra ventaja de su uso es que permite controlar la información que circula a través de cada reflector, pudiendo manipular dicha información en función de circunstancias determinadas para, por ejemplo, modificar o encriptar su codificación, o bien detener la retransmisión de algún medio [119].

Por el contrario, es una alternativa que limita la escalabilidad debido a que toda la responsabilidad del control del servicio, del envío de tráfico desde el grupo multicast hacia el exterior y de la recepción del tráfico exterior y posterior difusión al seno del grupo recae en los reflectores. Una variante que trata de paliar este inconveniente consiste en soluciones híbridas, donde los propios miembros del grupo multicast pueden ejercer en un momento dado el rol de reflectores, y así repartir la carga que soporta la overlay [133].

3.2.3. Discusión

	ALM P2P	ALM no P2P	Reflectores
Latencia	-		+
Escalabilidad	+		-
Robustez	+		-

Figura 3.4: Comparativa entre las ALM y los reflectores.

La figura 3.4 ilustra una comparativa de las características de cada tipo de overlay. Las ALM garantizan una mayor escalabilidad que las overlays compuestas por reflectores. El número de miembros que se pueden interconectar en una ALM es más elevado porque el hecho de añadir un miembro no supone una

sobrecarga a la overlay. La inclusión de un nuevo miembro tan solo compromete a los miembros que pasan a ser adyacentes del que ha sido añadido. Por su parte, en una overlay de reflectores, al añadir un nuevo reflector se aumenta el tráfico que envían y reciben los restantes reflectores. Otra diferencia con respecto a los reflectores es que las ALM son overlays más flexibles y robustas, ya que eliminan el riesgo de dependencia directa de un elemento concreto, como es el caso del reflector. Por el contrario, las ALM tienen el inconveniente de originar una mayor latencia en la entrega de la información puesto que las rutas de transmisión son notablemente más largas, por lo que no son adecuadas para comunicaciones en tiempo real interactivas bajo severas restricciones en cuanto al retardo máximo admisible. Por tanto, dada su idoneidad en términos de latencia, la técnica de reflectores unicast/multicast fue la utilizada en la plataforma de comunicación original sobre la que opera la herramienta e-pSyLon [40].

Capítulo 4

Discusión de resultados

En este capítulo se presentan los resultados más destacados obtenidos durante la tesis, los cuales están divididos en tres secciones. En primer lugar, en la sección 4.1 se describen los resultados relacionados con la investigación sobre la literatura relacionada con el paradigma de la computación autónoma (CA), y en particular dentro del ámbito de los sistemas de comunicación multimedia (SCM), que constituyen la base para el desarrollo de la tesis. A continuación, en la sección 4.2 se traslada el conocimiento adquirido de la investigación anterior al caso concreto de la plataforma de distribución sobre la que opera la herramienta e-pSyLon. Como resultado, se presenta el diseño y evaluación de una plataforma destinada al e-learning síncrono que incorpora parte de las características autónomas identificadas durante el análisis de la literatura. Finalmente, en la sección 4.3 se presentan los resultados relacionados con la investigación sobre el e-learning síncrono, tanto evaluando la calidad de la experiencia de usuario utilizando la plataforma resultante del diseño de la sección 4.2, como sobre las nuevas tendencias que han aparecido dentro del e-learning síncrono.

4.1. Investigación sobre sistemas autónomos

Como se ha visto anteriormente, las propiedades que componen inicialmente el paradigma de los sistemas autónomos son cuatro: auto-configuración, auto-optimización, auto-curación y auto-protección. Sin embargo, desde un principio se postuló que probablemente el número de propiedades iría aumentando a medida que se intensificasen las investigaciones al respecto [111]. En los últimos años el abanico de propiedades autónomas se ha ampliado notablemente [13, 30, 110], e incluso se han enunciado relaciones de interdependencia entre algunas de ellas [13]. Debido a que un sistema autónomo es todo aquel que es capaz de auto-gestionarse, el paradigma de los sistemas autónomos puede aplicarse en muchas parcelas de la ciencia y la tecnología [30]. En la presente tesis se ha realizado un estudio sobre el paradigma de la CA dentro del ámbito de los SCM, el cual se encuentra reflejado en la publicación número 1 del compendio de publicaciones (sec. 6.1.1) [87]. Más concretamente, el estudio se centra en las

overlays de distribución de datos sobre las que operan tales sistemas, y se puede dividir en las tres fases siguientes: identificación de subpropiedades autonómicas aplicables a los SCM, implementación de tales subpropiedades en los SCM y clasificación de las técnicas de implementación.

4.1.1. Identificación de subpropiedades autonómicas aplicables a los SCM

En la primera fase se ha analizado la literatura científica tratando de identificar todas aquellas propiedades autonómicas que pudiesen ser extrapolables, y por tanto ser observadas, en los SCM. A medida que el estudio descubre nuevas propiedades se considera que estas difieren sustancialmente de las propiedades fundamentales de los sistemas autonómicos y que resulta más sencillo definir relaciones más específicas entre ellas, y por tanto, establecer sus límites. Así pues, se define a estas nuevas propiedades como subpropiedades, ya que muestran parte de los comportamientos propuestos para las propiedades autonómicas iniciales. Concretamente, se han enunciado tanto la expresión algebraica como el lema siguientes para definir el concepto de subpropiedad autonómica. Sea S^* el conjunto clásico de las cuatro propiedades fundamentales de los sistemas autonómicos y Q el conjunto de subpropiedades identificadas en la literatura científica extrapolables a los sistemas de comunicación multimedia, entonces:

$$S^* = \{SC, SH, SO, SP\}$$

$$\forall s_i \in S^* \Rightarrow \exists Q' \subset Q / s_i \longrightarrow Q'$$

Lema. *Una subpropiedad autonómica, de manera individual o combinada con otras subpropiedades, da lugar a un comportamiento asociable con el de una de las cuatro propiedades autonómicas fundamentales.*

A continuación se presentan las subpropiedades que han sido identificadas durante el estudio y también sus relaciones con las cuatro propiedades fundamentales de los sistemas autonómicos. Además, dichas relaciones se ilustran de manera gráfica en la figura 4.1.

Auto-organización: es la capacidad de un sistema para interactuar con el entorno y con otros sistemas para así satisfacer objetivos que superan su capacidad, todo ello sin ser supervisado ni administrado por una entidad externa al sistema [74]. Inicialmente se plantea como una subpropiedad relacionada con la auto-configuración. No obstante, ya que el objetivo de modificar la organización de un sistema puede ser mejorar, restablecer o mantener ciertas características [109], la auto-organización da lugar a optimizar, recuperar o proteger el sistema, por lo que esta subpropiedad también está relacionada con las tres restantes propiedades autonómicas. Es una subpropiedad muy frecuente

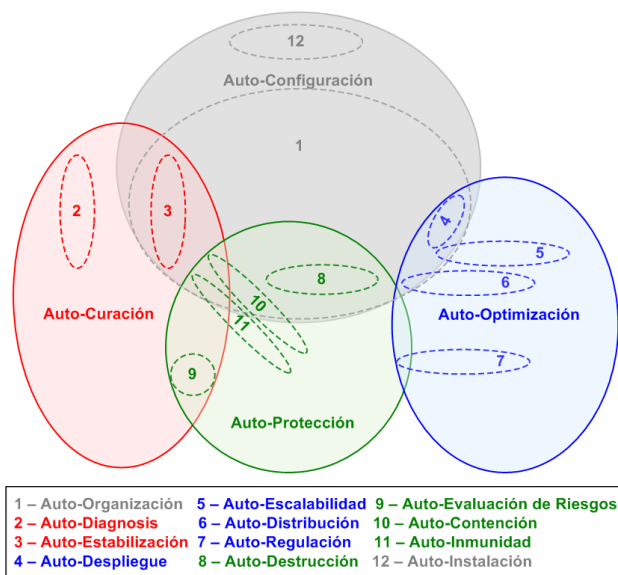


Figura 4.1: Subpropiedades autónomas y su relación con las propiedades fundamentales.

en los SCM puesto que los miembros de una overlay modifican las conexiones entre ellos, tanto para mejorar su rendimiento como para solventar problemas en los canales de comunicación.

Auto-diagnos: se refiere a la capacidad de un sistema para detectar un funcionamiento inadecuado, así como localizar e identificar la causa del mismo [49]. Para poder restaurar el funcionamiento apropiado de un sistema es necesario diagnosticar de manera precisa cuál es la causa del error, por tanto, todo sistema auto-curable debe ser auto-diagnosticable [105]. Esta subpropiedad también se encuentra ampliamente implementada en las overlays de los SCM, ya que sus miembros son capaces de detectar interrupciones en los canales de comunicación.

Auto-estabilización: también conocida como auto-recuperación, es la respuesta automática de un sistema ante errores para alcanzar un estado de equilibrio [31]. Esta subpropiedad es complementaria a la auto-diagnos, puesto que las acciones que lleva a cabo se realizan tras un diagnóstico apropiado. En los SCM esta subpropiedad está muy relacionada con la auto-organización, ya que los miembros de una overlay establecen nuevas comunicaciones entre ellos para solventar interrupciones en los canales de comunicación.

Auto-despliegue: es la capacidad de un sistema para construir relaciones entre sus miembros de manera incremental para alcanzar un régimen operativo que satisfice un objetivo de despliegue [57]. Para algunos autores el auto-despliegue es un caso especial de auto-estabilización [36], por tanto, se puede asociar con la auto-organización. Además, también tiene una componente auto-optimizadora puesto que el auto-despliegue busca obtener una organización inicial del sistema lo más eficiente posible. En los SCM esta subpropiedad encapsula la capacidad de un sistema de manejar la unión de nuevos miembros y establecer su ubicación inicial en la overlay.

Auto-escalabilidad: se refiere a la adaptación adecuada de un sistema a la demanda que se le requiere por parte de usuarios u otros sistemas [34]. En los SCM la auto-escalabilidad se refiere a la reacción de un sistema para no comprometer su rendimiento a medida que aumenta el número de miembros de la overlay. Por tanto, esta subpropiedad está relacionada con la auto-optimización [75], pero también con la auto-organización, ya que para favorecer la escalabilidad puede ser necesario realizar variaciones en la organización de la overlay [35].

Auto-distribución: es la capacidad de un sistema para determinar la disponibilidad de sus componentes, planificar y balancear las cargas de trabajo, así como asignar y coordinar recursos de acuerdo a las demandas del sistema [88]. La coordinación automática de recursos y cargas de trabajo son aspectos sobradamente relacionados con la auto-optimización [38]. No obstante, la auto-distribución también puede forzar cambios en la organización del sistema. Esto es apreciable en SCM donde las fuentes de datos pueden modificar su régimen de actuación para combinar su capacidad. También en aquellos SCM donde existen miembros que pueden convertirse en fuentes de datos bajo determinadas circunstancias.

Auto-regulación: es la capacidad de un sistema para ajustar sus parámetros de funcionamiento para así garantizar la calidad de servicio en función de circunstancias operacionales o de su entorno [81]. La auto-regulación está relacionada con la auto-optimización, asegurando aspectos como ciertos niveles de servicio o rendimiento [68]. En ocasiones, el hecho de mantener un cierto nivel de servicio sirve para proteger a un sistema, ya que disminuye la aparición de errores en sus componentes debido al exceso de utilización. En los SCM la auto-regulación puede observarse en aquellos sistemas capaces de modificar sobre la marcha el número, la codificación o la encriptación de los medios que distribuye. Con todo ello, la auto-regulación ayuda a incrementar la seguridad de los SCM previniéndolos de usos malintencionados [24].

Auto-destrucción: se refiere a la capacidad de un sistema para detener su ejecución cuando detecta un problema o comportamiento irreversible que compromete su seguridad o la de su entorno [110]. Esta propiedad se pone de manifiesto en los SCM especialmente cuando se retransmiten medios que

consumen muchos recursos, ya que la detención de un medio, por ejemplo sacrificar el vídeo para mantener el audio, puede evitar saturaciones en la overlay. Aunque esta subpropiedad está claramente relacionada con la auto-protección, también guarda relación con la auto-organización. Por ejemplo, se observa en aquellas ocasiones en las que para proteger al sistema puede ser necesario expulsar de la overlay a uno de sus miembros, o no permitir el acceso de nuevos miembros.

Auto-evaluación de riesgos: representa la habilidad de un sistema para predecir y evaluar vulnerabilidades, agujeros de seguridad, o interrupciones de servicio. El sistema no solo es capaz de predecir un riesgo concreto, sino también el efecto, las reacciones en cascada o las futuras implicaciones de las acciones realizadas para mitigar tal riesgo [11]. Una vez valorado el riesgo, un sistema autónomo puede actuar de manera proactiva o reactiva, lo que guarda implicaciones con la auto-protección y la auto-curación respectivamente [21]. En el contexto de los SCM, esta propiedad está presente en los sistemas donde sus miembros son capaces de estimar el impacto de una circunstancia operacional tanto en el rendimiento de la overlay como en la calidad de servicio observada por los usuarios.

Auto-contención: se puede encontrar en los sistemas capaces de minimizar el impacto de acciones malintencionadas y fallos, afectando solo a una parte del mismo, mientras que la operatividad se mantiene con la parte no comprometida del sistema [13]. La auto-contención está principalmente relacionada con la auto-protección. Sin embargo, la capacidad de restaurar la operatividad del sistema establece un vínculo con la auto-curación. Además, en los SCM la auto-contención puede apoyarse en la auto-organización para alcanzar su objetivo, por ejemplo, para aislar a miembros que comprometen el rendimiento o la funcionalidad de la overlay.

Auto-inmunidad: es la capacidad de un sistema para restaurar la seguridad y la robustez tras la primera aparición de un evento nocivo, a raíz del cual el sistema será capaz de adaptarse para que tal evento no vuelva a comprometer al sistema. Dos lemas se proponen en [13] para definir la relación entre auto-inmunidad, auto-curación y auto-protección:

- Todo sistema auto-inmune es auto-curable, pero lo contrario no es cierto.
- Todo sistema auto-protegido es auto-inmune, pero lo contrario no es cierto.

En el ámbito de los SCM la auto-inmunidad puede observarse en aquellos sistemas donde sus miembros son ubicados en la overlay en función de la probabilidad de causar particiones en la misma, por ejemplo, ubicando a los miembros potencialmente peligrosos en los márgenes de la overlay.

Auto-instalación: es la capacidad de un sistema para añadir nuevos componentes y software, o reinstalar programas o módulos tras errores o degradaciones [110]. Esta característica guarda relación con la auto-configuración. No es una subpropiedad muy extendida en los SCM, pero puede observarse en aquellos donde sus miembros obtienen codecs durante la fase de despliegue, o si sus miembros pueden cambiar su funcionamiento sobre la marcha copiando el comportamiento o estrategia funcional de otros miembros de la overlay.

4.1.2. Implementación de subpropiedades autonómicas en los SCM

Del estudio anterior se desprende que la auto-gestión es una parte inherente de los SCM y, en especial, de sus overlays. Por tanto, sirven como ejemplos interesantes y representativos de los sistemas autonómicos. En la segunda fase, para profundizar más en el estudio, se han analizado los trabajos científicos más relevantes sobre los SCM, identificando en cada uno qué subpropiedades se encuentran implementadas.

Se han estudiado un total de 30 trabajos científicos. Además, se incluye la versión de la plataforma de distribución para la herramienta e-pSyLon que se ha desarrollado durante esta tesis, y que se explicará en detalle en la sección 4.2. La tabla 4.1 muestra la implementación de cada subpropiedad autonómica en las publicaciones analizadas dentro del ámbito de los SCM.

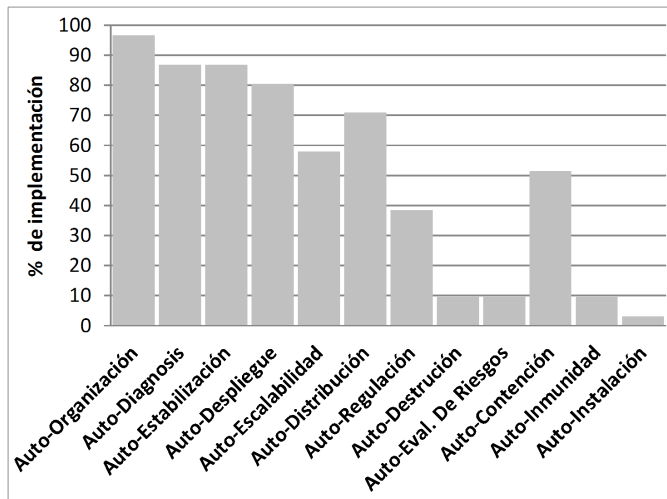


Figura 4.2: Porcentaje de implementación de cada subpropiedad autonómica.

Además, en la figura 4.2 se representa el porcentaje de implementación de cada subpropiedad en los SCM, tomando como referencia el total de trabajos analizados. Aunque la muestra puede considerarse poco significativa debido a su tamaño, sí resulta representativa debido a la importancia y la relevancia de los trabajos que la componen. Puede observarse cómo las subpropiedades relacionadas con la auto-configuración y la auto-optimización son las más implementadas en este tipo de sistemas.

Tabla 4.1: Subpropiedades autónomas en los SCM.

Trabajo Científico	Auto-Organización	Auto-Diagnosis	Auto-Estabilización	Auto-Despliegue	Auto-Escalabilidad	Auto-Distribución	Auto-Regulación	Auto-Destrucción	Auto-Eval. Riesgos	Auto-Contención	Auto-Inmunidad	Auto-Instalación
[Chawathe et al. 2000] [20]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Jannotti et al. 2000] [63]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Pendarakis et al. 2001] [92]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Roca and El-Sayed 2001] [98]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
[Banerjee et al. 2002] [8]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Chu et al. 2002] [22]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Castro et al. 2003] [19]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Hefeeda et al. 2003] [51]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Luo et al. 2004] [73]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Tran et al. 2004] [114]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Liu et al. 2005] [69]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Zhang et al. 2005] [135]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Banerjee et al. 2006] [9]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Hales and Arteconi 2006] [46]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
[Song et al. 2006] [108]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Venkataraman et al. 2006] [118]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Baccichet et al. 2007] [7]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Maraviglia et al. 2007] [76]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Pianese et al. 2007] [94]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Zhu et al. 2007] [137]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Mol et al. 2008] [80]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Yin et al. 2008] [130]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
[Liu et al. 2009] [70]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Bikfalvi et al. 2010] [14]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Ding et al. 2010] [27]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Huang et al. 2010] [58]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Payberah et al. 2010] [91]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Qiu et al. 2010] [96]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Wang et al. 2010] [123]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
[Wang and Chen 2011] [124]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[Granda et al. 2013] [43]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4.1.3. Taxonomía para clasificar las implementaciones de las subpropiedades autonómicas

Los SCM analizados implementan subpropiedades autonómicas para administrar sus overlays. Las entidades que componen una overlay pueden actuar de manera distribuida e independiente, sin necesidad de un conocimiento exhaustivo acerca del sistema completo, de tal manera que pueden gestionar la overlay por sí mismos, sin que dicha gestión implique una degradación en la calidad de servicio experimentada por los usuarios. Los SCM implementan cada una de las subpropiedades mediante técnicas de auto-gestión que permiten afrontar situaciones como la reorganización de la overlay, ya sea para aumentar su rendimiento, garantizar la calidad de servicio o solventar errores en los miembros que componen la overlay. Sin embargo, una subpropiedad autonómica puede llevarse a cabo mediante varias técnicas.

Las diferencias entre técnicas surgen a raíz de las características de cada SCM, así como de su propósito. Por ejemplo, los SCM destinados a realizar comunicaciones entre grupos con un número pequeño de participantes difieren sustancialmente de aquellos destinados a realizar comunicaciones entre un usuario y muchos receptores, o entre muchos usuarios concurrentemente. También surgen diferencias entre los SCM en función de requisitos como la máxima latencia admisible. Estas diferencias pueden observarse en aspectos como una administración centralizada o distribuida de la overlay, una topología de la overlay de tipo malla o árbol, o el enfoque de distribución de los datos.

En la tercera fase de la investigación se han analizado los trabajos recopilados en la tabla 4.1, identificando las diferentes técnicas de implementación que dan lugar a una subpropiedad autonómica. En vista de las diferencias apreciadas entre las distintas técnicas para obtener un mismo propósito, se ha propuesto la taxonomía ilustrada en la figura 4.3 para poder categorizar, en base a los criterios que la componen, cada una de las técnicas que implementan una subpropiedad autonómica. Dicha taxonomía es, dentro de nuestro conocimiento, la primera que categoriza las técnicas que implementan subpropiedades autonómicas. No obstante, en la literatura científica existen taxonomías que han sido propuestas haciendo referencia a diferentes aspectos relacionados con la CA tanto en términos generales [113], como en entornos de computación grid [97], analizando la naturaleza de las propiedades fundamentales de la CA [25], o estudiando la auto-organización dentro de una familia muy concreta de overlays [4]. La explicación detallada de los criterios de la taxonomía, así como su aplicación para clasificar cada una de las técnicas identificadas, pueden consultarse en la publicación número 1 del compendio de publicaciones (sec. 6.1.1).

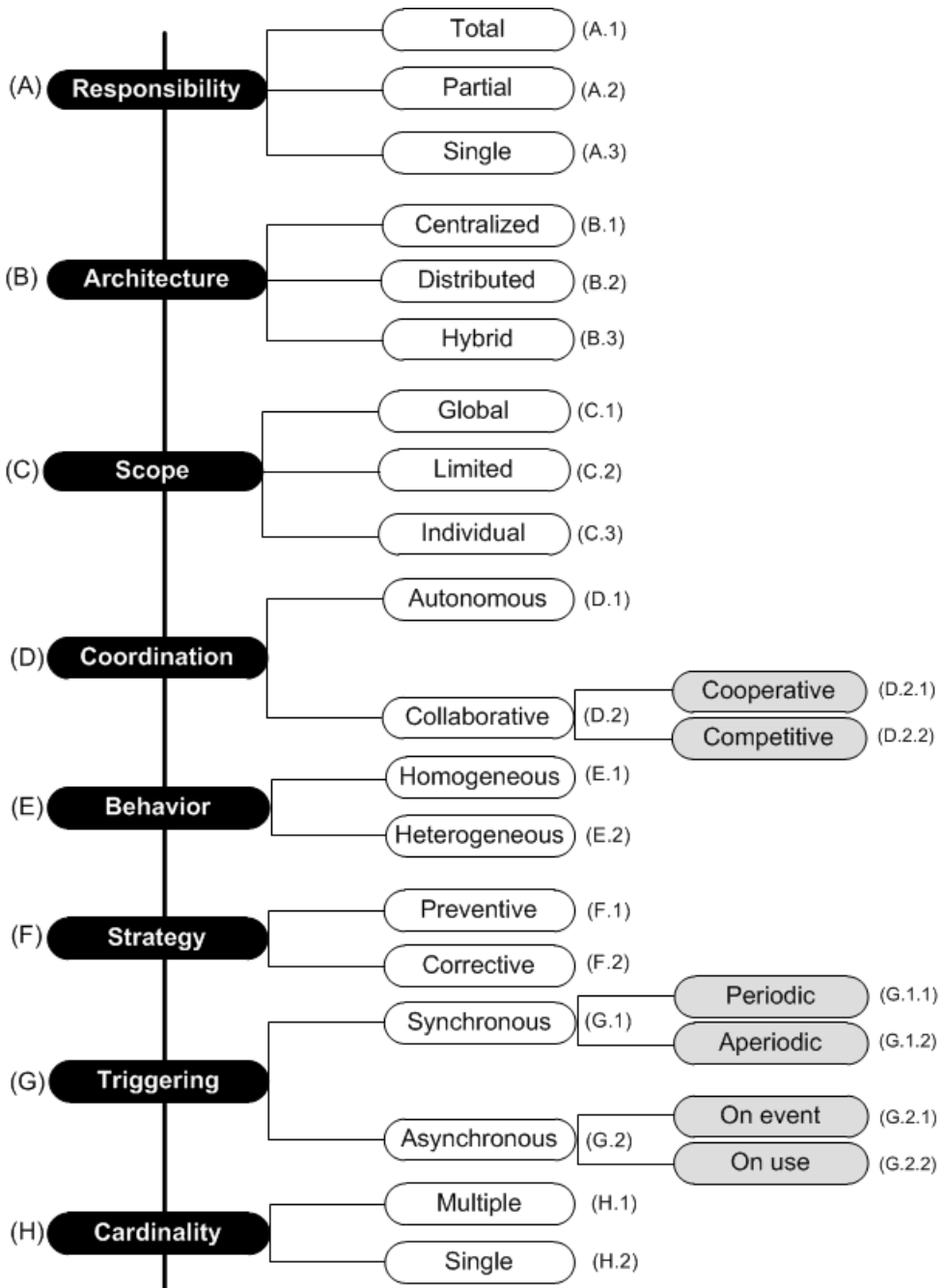


Figura 4.3: Taxonomía para clasificar las técnicas self-* en los SCM.

4.2. Diseño e implementación de la plataforma de distribución

En este apartado se presentan el diseño e implementación de la plataforma de distribución multimedia sobre la que opera la herramienta e-pSyLon, destinada a la realización de actividades de e-learning síncrono tanto en el ámbito académico como corporativo. Concretamente, el trabajo desarrollado ha consistido en ampliar el funcionamiento de la plataforma de distribución inicial [40]. Dicha plataforma era muy básica ya que adolecía de varias características autonómicas que han sido incorporadas durante la realización de esta tesis doctoral.

En primer lugar, se aborda el diseño de la overlay sobre la que la plataforma distribuye la información multimedia. En segundo lugar, se profundiza en las características autonómicas extraídas del análisis presentado en el apartado 4.1 que han sido implementadas en la plataforma. Por último, se muestran una serie de resultados que ilustran el rendimiento de la plataforma. El trabajo resumido a lo largo de este apartado se encuentra desarrollado en las publicaciones número 2, 3, 4, 5 y 8 del compendio de publicaciones (secciones 6.1.2,..., 6.1.5 y 6.2.1).

4.2.1. Diseño de la overlay de la plataforma de distribución

La overlay de la plataforma de distribución está diseñada basándose en tres redes virtuales tal y como se ilustra en la figura 4.4. En primer lugar, una red virtual de reflectores es la responsable de la distribución de la información multimedia entre los participantes que tomen parte en las actividades de e-learning síncrono. En segundo lugar, una red virtual de señalización proporciona a los participantes el acceso a la actividad y facilita la negociación de la configuración de la información multimedia que se retransmite durante las actividades. Finalmente, una red virtual de control es utilizada para reorganizar la red de reflectores en función de la unión y abandono de participantes o de errores en la red subyacente, de tal manera que se asegura la eficiencia en la entrega de la información multimedia en tiempo real.

La principal ventaja del diseño de la overlay es su modularidad, que facilita la flexibilidad y la portabilidad de la plataforma, ya que cualquiera de las redes virtuales puede ser modificada en su diseño o topología sin afectar al resto de redes virtuales. La gestión y administración de estas redes virtuales se realiza de manera centralizada mediante un elemento de control denominado comúnmente en la literatura como *Rendezvous Point* (RP) [33]. El diseño de la overlay, así como la evaluación del comportamiento de la misma, fue reflejado en la publicación número 3 del compendio de publicaciones (sec. 6.1.3).

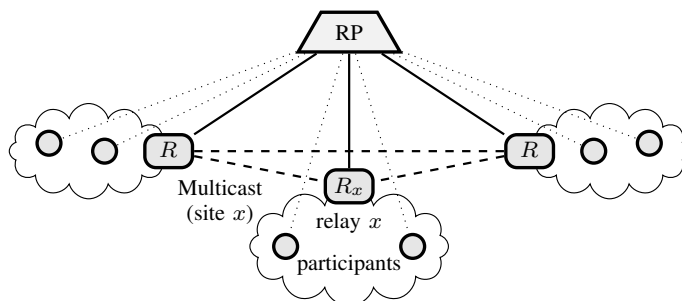


Figura 4.4: Overlay de la plataforma de distribución de la herramienta e-pSyLon.

4.2.1.1. Red virtual de reflectores

La infraestructura de red de una gran corporación puede considerarse como un grupo de islas multicast conectadas mediante una red de comunicaciones sin capacidad multicast. El uso de reflectores permite solventar tal circunstancia de tal manera que el uso de varios reflectores interconectados entre sí da lugar a una red virtual donde cada reflector ejerce la función de proxy entre una isla multicast y el resto. Como se aprecia en la figura 4.4 mediante líneas punteadas gruesas, todos los reflectores están conectados mediante una topología *full-mesh*. Esta topología minimiza el retardo en las comunicaciones ya que garantiza que los datos que emite un participante circulan por un máximo de dos elementos antes de alcanzar su destino.

La red virtual de reflectores es utilizada para distribuir la información multimedia que se genera durante la actividad. El protocolo RTP (*Real-time Transport Protocol*) [102] está considerado el estándar para la distribución de medios continuos como el audio o el vídeo sobre redes IP, aunque también puede utilizarse para transportar otros medios con restricciones temporales. Toda la información multimedia generada durante las actividades de e-learning se retransmite mediante el protocolo RTP. Por tanto, la función de cada reflector es replicar el tráfico entrante RTP hacia los participantes de la actividad o hacia otros reflectores. El reflector determina para cada paquete RTP y RTCP¹ recibido cuál es el destino del mismo y lo retransmite hacia el interior de la isla multicast, hacia la red virtual de reflectores, o hacia los participantes que puedan existir fuera del ámbito multicast.

¹El protocolo RTCP (*RTP Control Protocol*) es el protocolo de control asociado a RTP. RTCP proporciona retroalimentación acerca de la calidad en la entrega de la información multimedia. Además, mediante el análisis del tráfico RTCP el reflector puede detectar errores ya que determina la presencia de participantes inactivos. Por tanto, el tráfico originado por este protocolo es muy relevante y debe ser retransmitido a través de la red virtual de reflectores junto con el tráfico RTP.

4.2.1.2. Red virtual de señalización

La red virtual de señalización utiliza el protocolo SIP (*Session Initiation Protocol*) [100] para la gestión de las actividades realizadas sobre la plataforma. SIP es un protocolo estándar de señalización que permite iniciar, administrar y terminar comunicaciones multimedia sobre redes IP. Además, permite negociar la configuración multimedia de los datos a retransmitir entre usuarios mediante el uso embebido del protocolo SDP (*Session Description Protocol*) [47]. Aunque inicialmente el protocolo SIP se diseñó para gestionar servicios de voz sobre IP (VoIP), los cuales involucran dos usuarios en las comunicaciones, puede ser utilizado de igual manera para gestionar conferencias entre múltiples usuarios [99].

Como se aprecia en la figura 4.4, mediante líneas punteadas de trazo fino, esta red virtual tiene una topología en forma de estrella. El RP ejerce el rol de núcleo central de la misma, lo que se denomina como foco SIP, administrando las conferencias entre participantes. Para ello, estos últimos establecen una comunicación no persistente denominada diálogo SIP con el RP, a través de la cual se gestiona para cada participante de manera individualizada su unión a la actividad, posibles redirecciones y su salida de la misma, así como la configuración multimedia de los datos a emitir y recibir.

4.2.1.3. Red virtual de control

La red virtual de control es, tal y como se ilustra en la figura 4.4 mediante líneas continuas, una red en forma de estrella donde el núcleo es el RP que a su vez está conectado con cada uno de los reflectores que componen la red virtual de reflectores. El RP utiliza el protocolo TCP [95] (*Transmission Control Protocol*) para establecer conexiones persistentes con los reflectores mediante las cuales notifica los cambios que se deben realizar para mantener optimizada la topología de la red virtual de reflectores. El contenido de los mensajes emitidos a través de esta red sigue el formato XML (*eXtensible Markup Language*) [17].

Los cambios en la topología de la red virtual de reflectores se deben principalmente a la necesidad de incluir o excluir reflectores de dicha red virtual en función de la presencia o ausencia de participantes en cada isla multicast. En otros casos, también puede ser necesario realizar cambios en la topología de la red virtual de reflectores en presencia de errores tanto en las entidades que la componen, como en la infraestructura de la red de comunicación subyacente.

Tabla 4.2: Estado de implementación de cada subpropiedad.

Subpropiedad	Implementada	Pendiente	No Aplicable
Auto-Organización	■	<input type="checkbox"/>	<input type="checkbox"/>
Auto-Diagnosís	■	<input type="checkbox"/>	<input type="checkbox"/>
Auto-Estabilización	■	<input type="checkbox"/>	<input type="checkbox"/>
Auto-Despliegue	■	<input type="checkbox"/>	<input type="checkbox"/>
Auto-Escalabilidad	<input type="checkbox"/>	<input type="checkbox"/>	■
Auto-Distribución	<input type="checkbox"/>	■	<input type="checkbox"/>
Auto-Regulación	■	<input type="checkbox"/>	<input type="checkbox"/>
Auto-Destrucción	<input type="checkbox"/>	■	<input type="checkbox"/>
Auto-Eval. Riesgos	<input type="checkbox"/>	■	<input type="checkbox"/>
Auto-Contención	<input type="checkbox"/>	■	<input type="checkbox"/>
Auto-Inmunidad	<input type="checkbox"/>	■	<input type="checkbox"/>
Auto-Instalación	<input type="checkbox"/>	■	<input type="checkbox"/>

4.2.2. Propiedades autónomas de la plataforma

El conjunto de subpropiedades autónomas obtenidas del análisis realizado en la sección 4.1.1 permite identificar qué subpropiedades pueden encontrarse implementadas en un sistema de comunicación multimedia. No obstante, una plataforma de e-learning síncrono tiene una serie de requisitos y restricciones impuestos por su carácter interactivo en tiempo real. El principal es asegurar una baja latencia y controlar el *jitter*, aunque otros aspectos como la fiabilidad son igualmente muy importantes. La implementación de cada subpropiedad vendrá marcada por su utilidad de cara a satisfacer las restricciones de este tipo de plataformas.

La tabla 4.2 resume el estado de implementación de cada subpropiedad en la plataforma de distribución de la herramienta e-pSyLon. Como se aprecia en la tabla, existen propiedades que aún no se han implementado. En los casos concretos de la auto-distribución y la auto-destrucción ambas serán abordadas como trabajo futuro. El resto de subpropiedades se incluirán a medida que crezca su implantación en otros sistemas de comunicación multimedia en tiempo real. Mención aparte merece la auto-escalabilidad. Como se ha comentado en la sección 3.2.2, para asegurar una latencia mínima las overlays de reflectores, como es el caso de la overlay original utilizada por la herramienta e-pSyLon, deben construirse conforme a una topología donde cada reflector se conecta con el resto (*full-mesh*), ya que así se limita a dos el número de elementos intermedios por los que circulan los datos entre emisor y receptor. La auto-escalabilidad es opuesta al uso de una topología de tipo *full-mesh*, ya que cada nueva isla multicast

implicará un aumento exponencial en el tráfico que circula entre reflectores. Por tanto, la auto-escalabilidad no tiene cabida en una plataforma de este tipo.

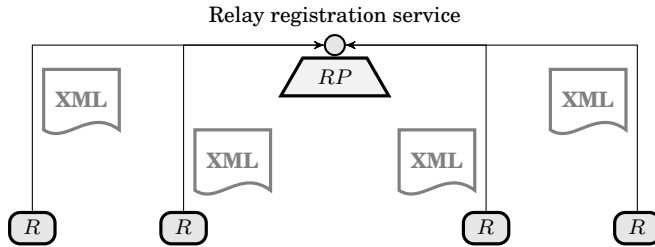


Figura 4.5: Primera fase de la técnica de auto-despliegue: registro de los reflectores.

Desde el punto de vista de las subpropiedades autonómicas presentes en la plataforma, las implementaciones han sido incluidas en las publicaciones número 2 [85] y 3 [43]. Concretamente, en la publicación número 3 se amplía la plataforma original de partida [40] con las subpropiedades de auto-despliegue, auto-organización, auto-diagnóstico, auto-estabilización y auto-regulación, las cuales están basadas en el uso de protocolos estándar como RTP y SIP.

La técnica de auto-despliegue se divide en tres fases. La primera fase, tal y como muestra la figura 4.5, consiste en el registro de reflectores sobre el RP. Mediante una conexión no persistente el reflector suministra al RP una serie de datos fundamentales para poder administrar cada una de las redes virtuales, como son: la dirección IP y puerto a utilizar por el reflector en la red virtual de control, las direcciones IP y puertos que el reflector usará en la red virtual de reflectores, y el identificador de la isla multicast en la que se encuentra. Los datos que los reflectores envían al RP se transmiten en formato XML. Al concluir esta fase, el RP dispondrá de información acerca de todos los reflectores, siendo capaz de construir la red virtual de control, si bien este hecho se pospone hasta que comience una actividad.

La segunda fase de auto-despliegue es un proceso ad-hoc para favorecer los beneficios del uso de una red corporativa donde la comunicación multicast está disponible. Esta fase se define como el proceso de identificación de la sede corporativa, y se ilustra en la figura 4.6. El reflector dispone de un servicio que atiende peticiones de participantes desde dentro de la isla multicast, de tal manera que cada participante al tratar de conectarse a la actividad envía antes una solicitud hacia la dirección multicast, la cual es atendida por el reflector, y de cuya respuesta el participante obtiene el identificador de la isla multicast.

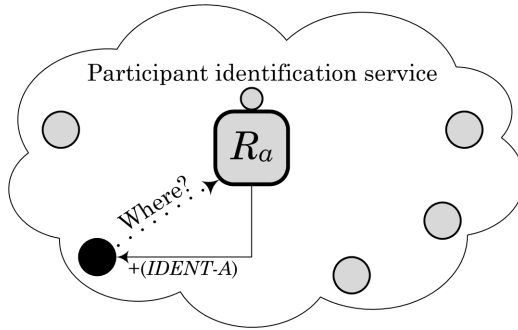


Figura 4.6: Segunda fase de la técnica de auto-despliegue: identificación de la sede corporativa.

En la última fase, que es común para todos los participantes de la actividad pertenezcan o no al ámbito de una isla multicast, se realiza el registro sobre el RP. Este proceso se ilustra en la figura 4.7. Cada participante dialoga con el RP utilizando el protocolo SIP. El RP asociará el identificador de la isla multicast suministrado por cada participante con alguno de los reflectores registrados. Si el participante no proporcionase ningún identificador (en caso de que no hubiese realizado la segunda fase por no pertenecer a ninguna isla multicast o esta hubiera sido fallida) el RP le asignará un reflector para que siga la actividad desde fuera del ámbito multicast.

Cabe destacar que en esta tercera fase es donde se realiza la auto-regulación de la plataforma puesto que, mediante el uso del protocolo SDP conjuntamente con el protocolo SIP, el RP y los participantes negocian la codificación de los formatos multimedia que se retransmitirán durante el transcurso de la actividad.

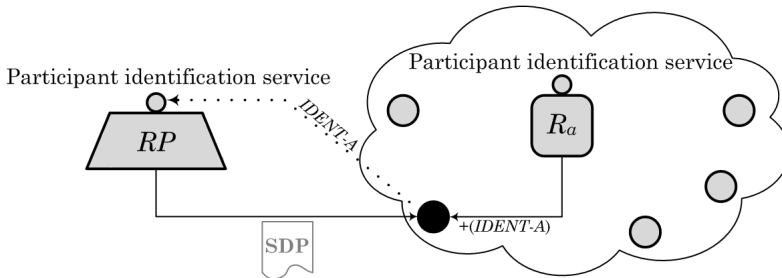


Figura 4.7: Tercera fase de la técnica de auto-despliegue: registro de los participantes.

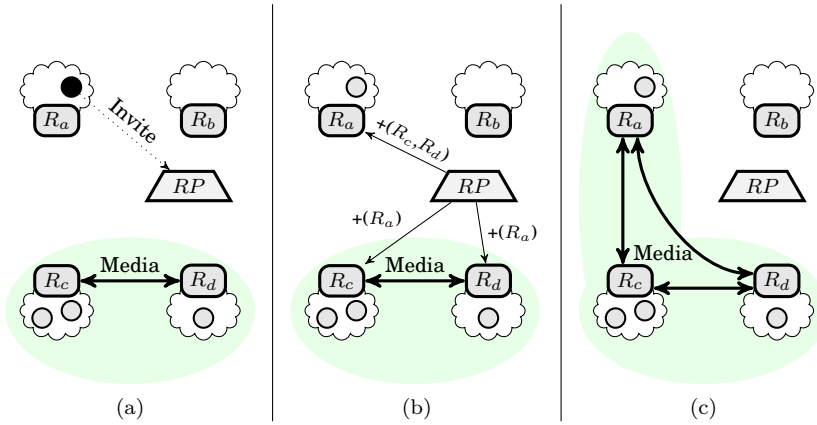


Figura 4.8: Inclusión de un reflector en la red virtual de reflectores.

Como consecuencia del anterior proceso de auto-despliegue, y más concretamente tras la primera fase, el RP dispone de información acerca de los reflectores y la ubicación de cada uno en las islas multicast. El proceso de auto-organización se basa en el uso de dicha información y se produce ante la ocurrencia de dos eventos concretos. En primer lugar, cuando el primer participante en una isla multicast trata de acceder a la actividad, es decir, cuando concluye con éxito la negociación de la tercera fase del proceso de auto-despliegue por parte de un participante desde una isla multicast que en ese momento no tiene participantes siguiendo la actividad. Cuando tal evento se produce, ilustrado en la figura 4.8, el RP detecta que es necesario modificar la red virtual de reflectores para dar cabida al nuevo participante en la actividad, por lo que el reflector ubicado en su isla multicast debe ser incluido en la red virtual de reflectores. Este hecho se lleva a cabo mediante el envío de mensajes a través de la red virtual de control desde el RP hacia cada uno de los reflectores que forman parte de la red virtual de reflectores y hacia el nuevo reflector.

El otro evento que implica la auto-organización de la red virtual de reflectores es el abandono de la actividad por parte del último participante ubicado en una isla multicast. Cuando ese hecho se produce, no es necesario seguir retransmitiendo tráfico hacia la isla multicast, por lo que la red virtual de reflectores debe ser actualizada para excluir su correspondiente reflector del conjunto de reflectores que la componen. Este proceso se muestra en la figura 4.9. Como se puede observar, es el RP el encargado de detectar la situación que requiere un cambio en la organización de la plataforma y actuar en consecuencia enviando los mensajes de control hacia el conjunto de reflectores que conforman en ese momento la red virtual de reflectores.

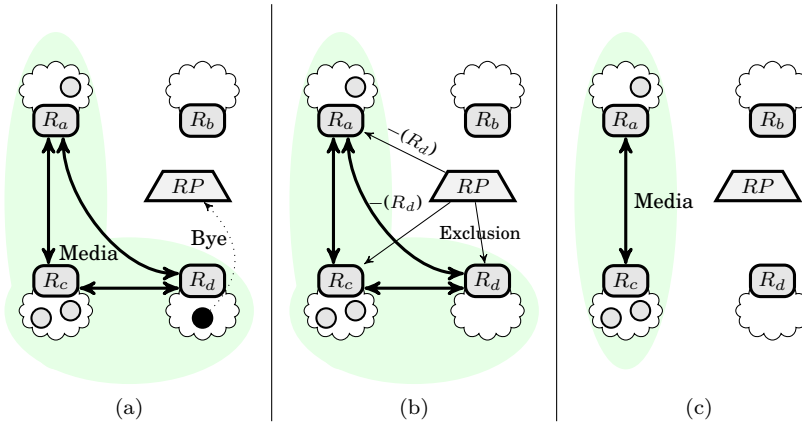


Figura 4.9: Exclusión de un reflector de la red virtual de reflectores.

Otras de las subpropiedades autónomas implementadas en la plataforma son la auto-diagnos y la auto-recuperación, que dan lugar a la capacidad de auto-curación de la plataforma. De esta manera es posible garantizar la estabilidad de las actividades en curso a pesar de errores en los elementos que componen la red virtual de reflectores. El proceso se resume de la manera siguiente. Cuando el RP pierde la conectividad con un reflector, debido a un error o un problema en la red subyacente, redirige a cada participante que administra el reflector fallido hacia algún otro de los que permanecen activos en la red virtual de reflectores. Esto se realiza mediante la construcción de enlaces unicast entre cada participante redirigido y el reflector que lo acoge. Así, los participantes pueden seguir tomando parte en las actividades en curso.

De manera más detallada, el proceso de auto-diagnos y auto-recuperación se ilustra en la figura 4.10. El RP hace uso de la red virtual de control, construida sobre el protocolo TCP, como mecanismo de detección de errores en la conectividad con los reflectores. Cuando una conexión TCP se termina abruptamente, el RP comienza el proceso de redirección para estabilizar los participantes que de otro modo no podrían seguir la actividad. Para ello determina, por cada participante a estabilizar, el reflector al que menos penalizaría su rendimiento individual, y el rendimiento de la plataforma en general, como resultado de acoger a un participante adicional mediante una conexión unicast. A partir de ese momento retoma el diálogo con el participante a través del protocolo SIP para negociar su redirección hacia el nuevo reflector, indicándole durante el diálogo la dirección IP del mismo. Este proceso de redirección puede ser utilizado para, de manera adicional, realizar una auto-regulación de la codificación multimedia de cara a acomodarse a las posibles limitaciones del reflector que acogerá al participante.

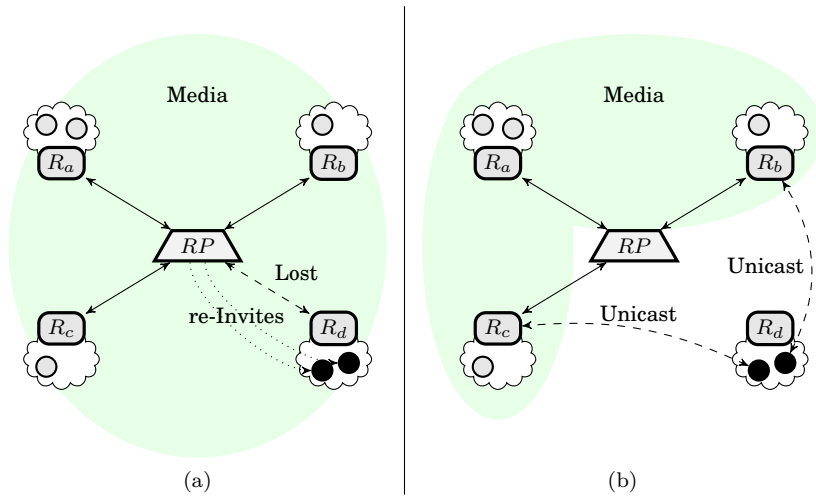


Figura 4.10: Técnica de auto-curación.

Cuando se reestablece la conectividad entre el RP y un reflector ya recuperado, se invoca el proceso de auto-organización para optimizar la red virtual de reflectores al punto de equilibrio previo al momento en el que se produjo el error del reflector. Dicho proceso consiste en redirigir de nuevo hacia el reflector recuperado a todos sus antiguos participantes, ya que reflector y participantes están ubicados en la misma isla multicast, y elimina la sobrecarga de utilizar enlaces unicast.

Finalmente, de manera paralela a la evolución de la plataforma, también se han estudiado aspectos como aumentar la seguridad mediante el uso del protocolo SRTP (*Secure Real-time Transport Protocol*) [10] (publicación número 4 [41]), o la fiabilidad mediante nuevas técnicas de auto-estabilización o mejoras sobre las ya existentes (publicación número 5 [42]). En la publicación número 8 [86] (en revisión) se han llevado a la práctica tales técnicas orientadas a la auto-estabilización, las cuales están basadas nuevamente en el protocolo SIP. La más destacada es la técnica que permite al RP recuperar el control de la actividad en curso cuando se producen errores que interrumpen su conectividad con las redes virtuales de control y señalización. En la figura 4.11 se muestra el pseudocódigo del proceso de recuperación del control de la actividad por parte del RP. A la conclusión de dicho proceso, el RP habrá reconstruido las dos redes virtuales mencionadas anteriormente. En dicha publicación se detalla también una técnica mejorada de auto-estabilización que afina el proceso de selección de un reflector como candidato a recibir participantes reubicados teniendo en cuenta los recursos de red disponibles en cada reflector.

```

procedure RECUPERARACTIVIDADENCURSO(actividad)
    ReflectoresActivos(actividad)  $\leftarrow$   $\emptyset$ 
    for all participante  $\in$  Participantes(actividad) do
        reflector  $\leftarrow$  ReflectorOriginal(actividad, participante)
        if ActivarReflector(actividad, reflector) then
            sdp  $\leftarrow$  CrearSDP(actividad, reflector)
            Reinvitar(participante, sdp)
            Asociar(actividad, reflector, participante)
            ReflectoresActivos(actividad)  $\leftarrow$  reflector
        else
            Redirigir(actividad, reflector, participante)
        end if
    end for
end procedure

procedure REDIRIGIR(actividad, reflector, participante)
    repeat
        nuevoReflector  $\leftarrow$  RedirigirA(Reflectores(actividad), reflector, participante)
    until ActivarReflector(actividad, nuevoReflector)
    sdp  $\leftarrow$  CrearSDP(actividad, nuevoReflector)
    Reinvitar(participante, sdp)
    Asociar(actividad, nuevoReflector, participante)
    ReflectoresActivos(actividad)  $\leftarrow$  nuevoReflector
    MarcarComoRedirigido(actividad, nuevoReflector, participante)
end procedure

```

Figura 4.11: Algoritmo de recuperación del control de la actividad.

4.2.3. Resultados de la plataforma

La eficiencia y comportamiento de la plataforma de distribución han sido evaluados mediante técnicas de simulación. Para ello, se ha utilizado el simulador de redes NS-3 [84]. El modelo construido simula una corporación dispersa en varias sedes, donde en el seno de cada una está disponible la comunicación multicast. Estas sedes se interconectan mediante una WAN (*Wide Area Network*) compuesta por enlaces de red de 20 Mbps con una latencia de 20 ms. La conexión del RP a través de la WAN se modela con un enlace de red de 10 Mbps y sin latencia adicional. El comportamiento de los usuarios se modela mediante la caracterización resumida en la tabla 4.3. Además, las actividades cuentan con un mecanismo de gestión de intervenciones que limita a cuatro el número de flujos, tanto de audio como de vídeo, que es posible retransmitir concurrentemente. Finalmente, los flujos de vídeo son generados usando el codec H.264 [125] con una tasa de bits variable y una resolución de 160×120 píxeles a 10 cuadros por segundo, mientras que los flujos de audio se generan usando el codec iLBC (*Internet Low Bit Rate Codec*) [5] con una tasa de bits constante y una carga útil de 20 ms. Los resultados expuestos a continuación se detallan en las publicaciones número 3 [43], y 8 [86] del compendio de publicaciones (secciones 6.1.3 y 6.2.1).

Tabla 4.3: Parámetros de la simulación a nivel de participante.

Tiempo de espera para unirse a la actividad (s)	λ_w	1/45
Probabilidad de abandono inesperado de la actividad	p_l	0.05
Intervalo de espera para la activación del audio (min)	(μ_a, σ_a)	(25,3.3)
Duración del flujo de audio (s)	(μ_{al}, σ_{al})	(15,2.6)
Intervalo de espera para la activación del vídeo (min)	(μ_v, σ_v)	(40,8.2)
Duración del flujo de vídeo (s)	(μ_{vl}, σ_{vl})	(30,9.5)
Intervalo de espera para la activación de las anotaciones (min)	(μ_w, σ_w)	(16.6,1.3)
Duración del flujo de anotaciones (s)	(μ_{wl}, σ_{wl})	(60,10)
Intervalo de espera para la activación del puntero virtual (min)	(μ_t, σ_t)	(20,1.6)
Duración del flujo de puntero virtual (s)	(μ_{tl}, σ_{tl})	(15,2.5)

En la figura 4.12 se muestra el comportamiento de la plataforma en relación al promedio del ancho de banda consumido en los reflectores en función de cada tipo de tráfico multimedia. En la figura 4.12a se mantiene constante el número de participantes y se varía el número de sedes, mientras que en la figura 4.12b se mantiene fijo el número de sedes mientras se varía el número de participantes. En ambos casos se puede comprobar el crecimiento brusco inicial del promedio de ancho de banda consumido, seguido de una relajación en dicho crecimiento que pasa a ser sostenido de manera asintótica.

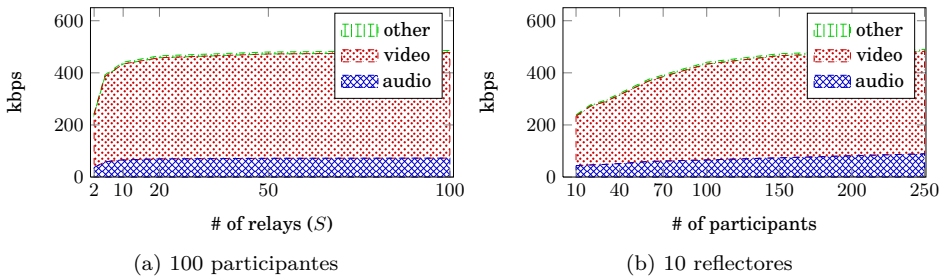


Figura 4.12: Promedio del ancho de banda consumido por cada tipo de tráfico en los reflectores.

El comportamiento anterior se debe a que, a medida que se van gestionando las intervenciones, se retransmiten más flujos multimedia a través de la plataforma, si bien al alcanzar el número máximo permitido de participantes concurrentes el tráfico multimedia ya no aumenta y tan solo lo hace el tráfico de control asociado al número de participantes de la actividad. Evaluando estos resultados con el modelo de tráfico teórico propuesto en la publicación número 3, se observa que el comportamiento de la plataforma es correcto, luego las técnicas

de auto-despliegue y auto-organización permiten construir una red virtual de reflectores que es eficiente desde el punto de vista del ancho de banda consumido.

En la figura 4.13 se representa el tiempo empleado por la técnica de auto-curación para recuperar todos los participantes de una sede en la que se simula el error de su reflector correspondiente. El tiempo total de recuperación engloba los tiempos parciales empleados por las técnicas de detección del error (auto-diagnos), de modificación de la de la red virtual de reflectores (auto-organización), de selección del reflector más apropiado como nueva ubicación para cada participante así como el posterior diálogo SIP entre el RP y los participantes para proceder a su redirección (auto-estabilización). Nuevamente, en la figura 4.13a se mantiene fijo el número de participantes variando el número de sedes, mientras que en la figura 4.13b se fija el número de sedes y se aumenta el número de participantes.

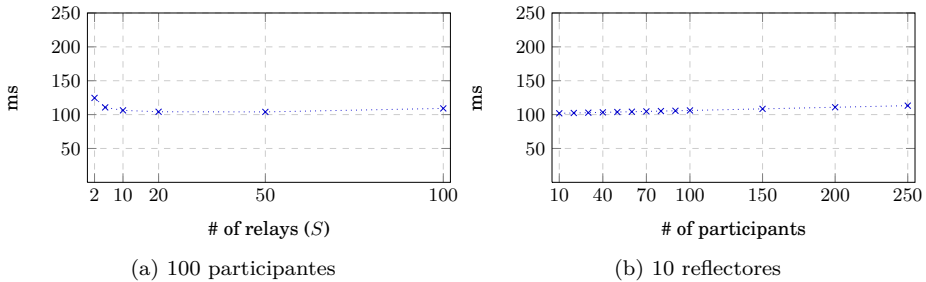


Figura 4.13: Tiempo empleado por la auto-curación para recuperar los participantes de una sede.

El tiempo para recuperar la plataforma tras el fallo de un reflector disminuye considerablemente a medida que el número de reflectores aumenta en actividades con un conjunto reducido de reflectores iniciales, tal y como se muestra en la figura 4.13a. Esto es debido a que al aumentar el número de sedes, disminuye el número de participantes a estabilizar. Este hecho deja de producirse cuando la dispersión de sedes es tan elevada que el proceso de auto-organización, donde el RP debe comunicarse con cada reflector, contrarresta la disminución del número de participantes por reflector, lo que origina un leve crecimiento en el tiempo de curación. Por su parte, el tiempo de curación resulta proporcional cuando se varía el número de participantes tal y como se representa en la figura 4.13b.

La figura 4.14 muestra el promedio de la pérdida de paquetes observada en los flujos de audio mientras se aplica la técnica de auto-curación. El promedio de paquetes perdidos consecutivos en un flujo de audio afectado directamente por el proceso de auto-curación es representado por una línea punteada de color

azul. Esos flujos se corresponden con aquellos recibidos en, o enviados desde, la sede donde está ubicado el reflector para el que se simuló su caída. El promedio de paquetes perdidos consecutivos observados respecto al conjunto de todos los flujos de audio que se intercambian en la plataforma se representa mediante una línea sólida de color rojo. Como en anteriores figuras, se varía el número de reflectores en el primer caso y el número de participantes en el segundo.

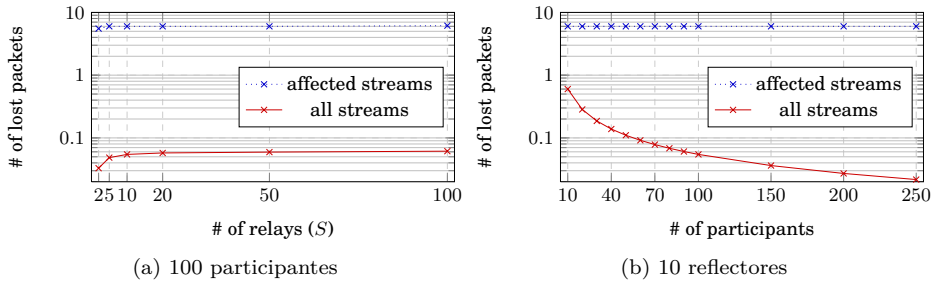


Figura 4.14: Paquetes de audio perdidos mientras se recuperan los participantes de una sede.

Los participantes tienen más probabilidades de seguir recibiendo el tráfico multimedia a través de IP multicast cuando ocurre el error de un reflector en actividades con un número bajo de reflectores, puesto que el número de participantes por sede es mayor. Así pues, los participantes de la sede no sufren pérdida de paquetes puesto que, aunque el reflector se caiga, seguirán recibiendo el tráfico originado en esa sede a través de la comunicación multicast. En los flujos de audio que son retransmitidos a través de la plataforma hacia otras sedes, o bien que son recibidos desde otras sedes, sí existe pérdida de paquetes. En el caso de los paquetes perdidos consecutivos en los flujos directamente afectados, el promedio aumenta ligeramente a medida que se incrementa el número de reflectores, tal y como se muestra en la figura 4.14a. Esto es consecuencia directa, como se observó en la figura 4.13a, del aumento del tiempo de curación a medida que se incrementa el número de reflectores cuando se trata de actividades con un elevado número de reflectores.

Respecto al promedio sobre el total de flujos de la plataforma, la pérdida de paquetes se incrementa más bruscamente según se va aumentando el número de reflectores en los casos donde el conjunto de reflectores iniciales es reducido. Esto es debido a que el hecho de aumentar el número de reflectores implica una mayor probabilidad de que algún emisor se encuentre en otra sede, por lo que un error en un reflector implica una mayor probabilidad de originar una pérdida de paquetes en flujos recibidos desde el exterior en los participantes de las restantes

sedes, lo que conlleva una mayor incidencia en el promedio total. Por último, al variar el número de participantes por sede, se puede apreciar cómo ese hecho no implica un aumento en el promedio del número de paquetes perdidos en los flujos directamente afectados a lo largo del proceso de reubicación de participantes en otro reflector. El promedio total de paquetes perdidos por flujo se diluye a medida que crece el número de participantes totales en la actividad, tal y como se aprecia en la figura 4.14b, ya que a mayor número de participantes aumenta el número de intervenciones y de flujos circulando por la plataforma, por lo que un error tiene menos incidencia en el promedio total.

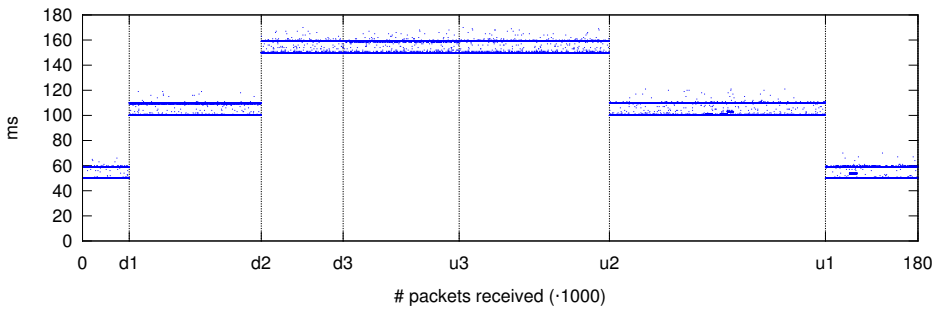


Figura 4.15: Evolución dinámica de la latencia

La latencia se ve incrementada durante el proceso de auto-curación puesto que parte de los participantes, e incluso el instructor, pueden ser reubicados teniendo que seguir la actividad a través de enlaces unicast, lo que implica un aumento del tráfico multimedia que circula por la plataforma. Por tanto, se ha determinado la penalización del proceso de auto-curación sobre la latencia en las comunicaciones en los participantes finales.

La figura 4.15 se corresponde con la latencia observada en un escenario en el que los paquetes se emiten desde un instructor, ubicado en la sede 1, hacia un participante p , ubicado en la sede 2, durante el transcurso de una actividad de 100 participantes repartidos de manera equitativa entre 20 sedes. Los eventos que se simulan para analizar la evolución de la latencia se detallan en la tabla 4.4. En el instante inicial, correspondiente con la figura 4.16a, la red virtual de reflectores se encuentra en un estado óptimo, puesto que no hay participantes reubicados. La latencia de los paquetes viene determinada exclusivamente por el retardo implícito de los canales de comunicación entre las sedes 1 y 2. Así pues, la información multimedia que genera el instructor es retransmitida desde el reflector R_1 hacia el R_2 , y desde este último hacia el participante p .

Tabla 4.4: Eventos simulados para analizar la evolución de la latencia.

d1	El reflector R_1 sufre una caída
d2	El reflector R_2 sufre una caída
d3	El reflector R_3 sufre una caída
u3	El reflector R_3 retorna de nuevo a la actividad
u2	El reflector R_2 retorna de nuevo a la actividad
u1	El reflector R_1 retorna de nuevo a la actividad

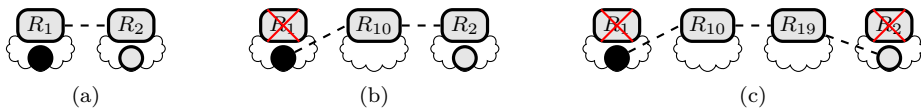


Figura 4.16: Estados de la red virtual de reflectores: (a) inicial; (b) tras evento d1; (c) tras evento d2.

Cuando el reflector R_1 sufre un error (evento d1), el instructor es reubicado hacia el reflector R_{10} , por lo que la información multimedia debe atravesar dos enlaces unicast hasta alcanzar su destino tal y como muestra la figura 4.16b, lo que origina un incremento en la latencia. La latencia aumenta todavía más cuando el reflector R_2 , que aloja al participante p , sufre un error (evento d2). En ese instante, el participante es reubicado en el reflector R_{19} , y la información multimedia generada por el instructor debe atravesar un enlace unicast adicional para alcanzar su destino como se representa en la figura 4.16c. En ese momento la plataforma se encuentra en el peor escenario posible, ya que la latencia en la comunicación entre instructor y participante supera el umbral de los 150 ms, que suele considerarse como el retardo máximo admisible en una comunicación de audio en tiempo real [93], donde la calidad de servicio experimentada por el participante puede verse afectada. No obstante, el impacto sobre el participante p es mínimo ya que la latencia sobrepasa ligeramente dicho límite, lo que supone una merma en la calidad, pero asumible pues se corresponde con el peor escenario posible. Los siguientes eventos (eventos d3 y u3) provocan un fallo y posterior vuelta a la actividad de un reflector que no penaliza la latencia en la comunicación entre instructor y participante. Cuando finalmente los reflectores del participante p y del instructor recobran la actividad (eventos u2 y u1), ambos son reubicados en sus sedes iniciales, por lo que vuelven a aprovechar la comunicación IP multicast. Por tanto, se elimina la necesidad de enlaces unicast adicionales y se minimiza la latencia en la comunicación entre ambos.

A continuación se muestran los resultados referentes a la comparación de la técnica mejorada de auto-estabilización, descrita en la publicación número 8 [86], respecto a la técnica de auto-estabilización de la publicación número

3 [43]. En estas pruebas se determina el número de participantes que pueden ser estabilizados. El procedimiento es simular la caída consecutiva de reflectores hasta que solamente quede uno operativo, y observar cómo la reubicación de los participantes entre los restantes reflectores afecta al ancho de banda disponible en cada reflector, y en qué momento alguno de ellos alcanza el punto de saturación. Si durante la recuperación de la sede *iésima*, al reubicar a un participante se alcanza la saturación de un reflector, el número total de participantes estabilizados será el que se hubiese acumulado durante la recuperación de las $i - 1$ sedes anteriores.

En estas pruebas se ha simulado el peor escenario posible desde el punto de vista de la sobrecarga de la plataforma, ya que todos los participantes envían audio de manera concurrente, es decir, sin limitación en el número de intervenciones en paralelo. Además, los recursos de red en los canales de comunicación entre reflectores no están equilibrados. Concretamente, el ancho de banda disponible en los canales de comunicación entre sedes se determina de acuerdo a la siguiente distribución normal $AB = \mathcal{N}(20, 2^2)$ en Mbps.

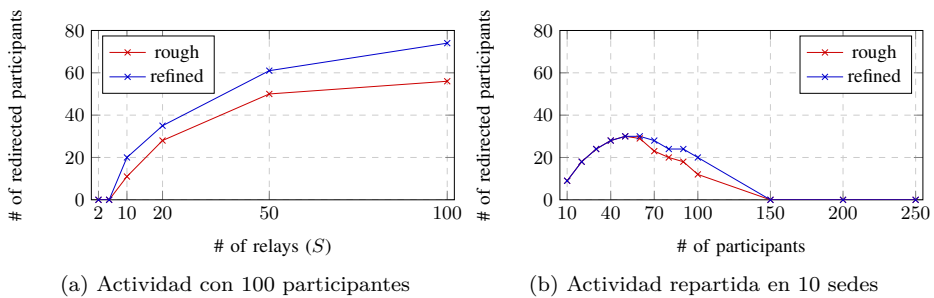


Figura 4.17: Número máximo de participantes estabilizados tras caídas consecutivas de reflectores.

La figura 4.17 representa el número máximo de participantes que pueden ser estabilizados. En la figura 4.17a se incrementa el número de reflectores manteniéndose fijo el número de participantes. En la figura 4.17b se muestra ese valor en función del incremento del número de participantes de la actividad. Ambas técnicas tienen un rendimiento semejante cuando el número de reflectores no es elevado. Sin embargo, a medida que aumenta el número de reflectores las diferencias comienzan a ser notables. Ese hecho se traduce en sedes completas que pueden ser recuperadas por la técnica mejorada, mientras que la versión anterior fracasa en ese mismo propósito. Desde el punto de vista de la variación de los participantes, el comportamiento es similar y las diferencias aparecen de

nuevo a medida que se incrementa el conjunto de participantes. No obstante, ambas técnicas fracasan al tratar de recuperar por completo la primera sede en la que se produce un error cuando el número de participantes es muy elevado.

La comparativa de la influencia en el rendimiento de la técnica de auto-estabilización en función de la variación en los recursos de red disponibles en los reflectores se representa en la figura 4.18. El escenario representa una actividad compuesta por 100 participantes y 10 sedes. En la figura 4.18a se muestra el número de sedes estabilizadas en función de la desviación estándar del ancho de banda de los reflectores. Una sede se considera estabilizada si todos sus participantes son estabilizados con éxito, es decir sin saturar ninguno de los reflectores restantes. Por su parte, la figura 4.18b representa el número de participantes estabilizados en función de la desviación estándar del ancho de banda de los reflectores. Como era de esperar, el rendimiento de la técnica mejorada es superior al de la versión previa a medida que crece la variación en el ancho de banda disponible entre los reflectores que componen la red virtual.

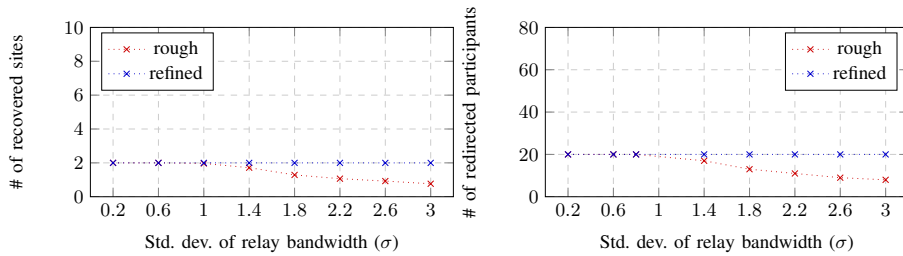


Figura 4.18: Número máximo de participantes e islas multicast estabilizados cuando los recursos de red no están balanceados.

Finalmente, la experimentación llevada a cabo para analizar la técnica empleada por el RP para recuperar el control de una actividad en curso se ilustra en la figura 4.19. La figura 4.19a se corresponde con el tiempo de recuperación en función del número de reflectores de la plataforma, mientras que la figura 4.19b se corresponde con el mismo tiempo en función del número de participantes. En el primer escenario se aprecia cómo el tiempo para recuperar el control de la actividad crece exponencialmente de acuerdo al número de reflectores. El comportamiento exponencial es debido a que el número de mensajes de control entre el RP y los reflectores crece exponencialmente, puesto que, tras cada reflector recuperado, se envía un mensaje de control indicando la dirección IP de dicho reflector a cada uno de los que componen la red virtual. Por otra parte, cuando la variación solo afecta al número de participantes, el tiempo

de recuperación crece linealmente puesto que solo varía el número de mensajes SIP que el RP debe enviar. Incluso en los casos de mayor sobrecarga el tiempo de recuperación no resulta demasiado elevado, siendo perfectamente asumible ya que este proceso se realiza sin interferir con la distribución de información multimedia a través de la red virtual de reflectores, por lo que en ningún momento se interrumpen las comunicaciones entre participantes. Por tanto, la técnica para la recuperación de las actividades en curso por parte del RP mediante el uso del protocolo SIP resulta razonablemente eficiente.

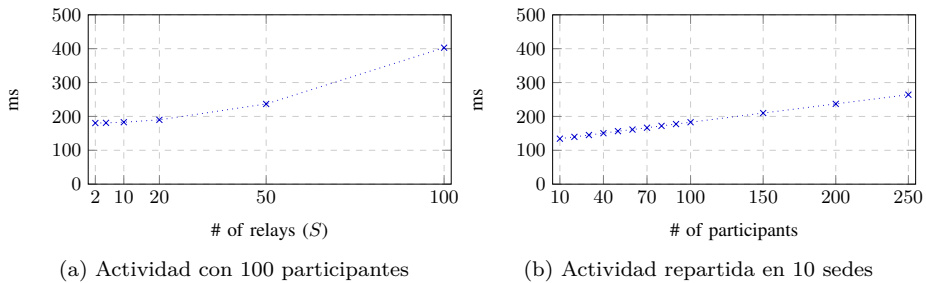


Figura 4.19: Tiempo empleado por el RP para recuperar el control de la actividad.

4.3. Investigación sobre el e-learning síncrono

En esta sección se describen los resultados relacionados con la investigación sobre el e-learning síncrono desde una perspectiva más orientada a la aplicación práctica. En primer lugar, se ha evaluado la calidad de la experiencia de usuario utilizando de manera conjunta la plataforma de distribución autónoma y la herramienta e-pSyLon durante actividades de e-training. El e-training es una variante del e-learning síncrono cuya única particularidad reside en el ámbito de aplicación ya que se define como el entrenamiento, formación o capacitación del personal de una empresa. En segundo lugar, se presentan las nuevas tendencias e innovaciones que se han identificado durante la investigación en la parcela del e-learning síncrono.

4.3.1. Resultados del uso de la plataforma autónoma y la herramienta e-pSyLon

A continuación se presentan los resultados de utilizar en un entorno corporativo la plataforma de distribución autónoma junto con la herramienta de e-learning síncrono e-pSyLon. El escenario de aplicación fue la corporación ArcelorMittal, líder mundial en la fabricación del acero. La temática desarrollada durante la actividad fue la realización de un curso sobre prevención de riesgos eléctricos. Las actividades se realizaron en una única jornada laboral y estaban planeadas para abarcar dos horas de duración. La edad media de los participantes fue de 33.5 años y la proporción entre hombres y mujeres de 7 a 1. Los resultados e información detallada sobre la experiencia se incluyen en la publicación número 6 del compendio de publicaciones (sec. 6.1.6) [45].

En la figura 4.20 se detallan las localizaciones geográficas de las sedes corporativas de ArcelorMittal que tomaron parte en la actividad. La distribución de participantes fue la siguiente: el instructor estaba situado en Avilés, un empleado en Gijón, tres en Lesaka y el resto en Sagunto. En la figura se muestra también el ancho de banda disponible en los enlaces de red de la WAN que interconecta las sedes corporativas, así como la latencia observada entre sedes, que fue calculada mediante la mitad del tiempo de ida y vuelta de un paquete.

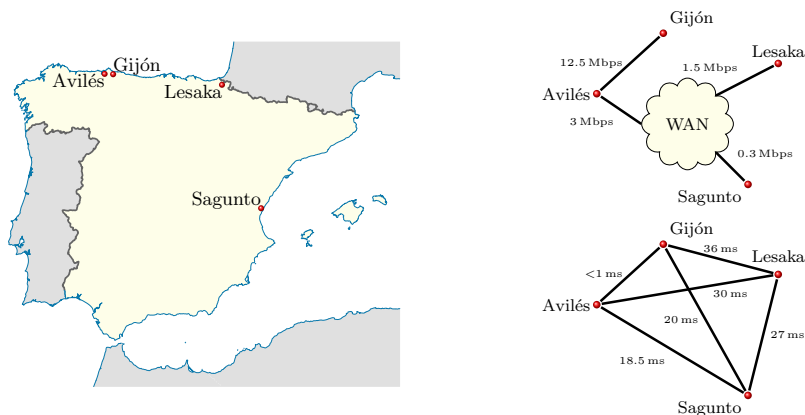


Figura 4.20: Localizaciones geográficas y ancho de banda disponible en el caso de uso de real.

El instructor ejercía también el rol de moderador, de tal manera que era el encargado de asignar los turnos de intervención de audio, vídeo, etc. Todos los participantes podían comunicarse con el resto utilizando las diversas funcionalidades multimedia que ofrece la herramienta e-pSyLon, excepto los

situados en la sede de Sagunto, ya que no estaban autorizados para enviar ni recibir información de vídeo. Dicha limitación fue impuesta debido a que, aunque el ancho de banda disponible en la sede de Sagunto (8 Mbps) es alto si se compara con el de las restantes sedes, el tráfico diario de las actividades realizadas en dicha sede implica que el ancho de banda real disponible sea muy limitado. Las intervenciones durante la actividad se limitaron a un máximo de dos participantes usando concurrentemente el vídeo, y tres participantes como máximo usando el audio. El códec de vídeo utilizado fue el VC-1 [64] con una resolución de 160×120 píxeles y 15 cuadros por segundo, lo que originaba un consumo de red de 106.96 kbps por cada flujo de vídeo. Por su parte, se usó el iLBC como códec de audio con una carga útil de 20 ms que implicaba un consumo de red de 38.4 kbps por cada flujo de audio.

La tabla 4.5 muestra las preguntas realizadas en el cuestionario final sobre la valoración de la herramienta e-pSyLon y la experiencia de aprendizaje global. En todas las preguntas se utiliza una escala Likert de cinco niveles para especificar el grado de acuerdo o desacuerdo. Las valoraciones en general fueron positivas, aunque los participantes notificaron algunas incidencias en referencia a la calidad del audio. Esas incidencias se deben a picos en el consumo de ancho de banda por el tráfico específico de las sedes, que comparte los enlaces de red con el generado durante la actividad, y no a problemas de eficiencia en la plataforma de distribución multimedia. No obstante, el número de incidencias no fue elevado y, en función de las respuestas de los participantes, se puede concluir que la herramienta e-pSyLon, junto con la plataforma de distribución de contenidos multimedia que dio soporte a la actividad, cumplieron satisfactoriamente con los requisitos relacionados con el desarrollo de actividades de e-learning síncrono.

Tabla 4.5: Evaluación de la calidad de la experiencia por parte de los participantes.

#	Cuestión	Respuesta
1	La actividad fue organizada adecuadamente	3.7
2	La duración de la actividad fue apropiada	3.5
3	La manera en la que se realizó la actividad facilita el aprendizaje	4.5
4	Todas mis dudas fueron resueltas	3.4
5	El instructor fomentó la participación durante la actividad	4.1
6	Estoy satisfecho con e-pSyLon al interactuar con otros participantes	3.9
7	Me gustaría utilizar la herramienta e-pSyLon en actividades futuras	4.1
8	Nivel de satisfacción general acerca de la experiencia de aprendizaje	4.0

4.3.2. Resultados sobre nuevos avances y tendencias en e-learning síncrono

A lo largo de esta fase de la investigación se han analizado varias herramientas de e-learning síncrono que han surgido desde la fecha en la que se presentó la herramienta e-pSyLon. Los resultados del estudio forman parte de la publicación número 7 del compendio de publicaciones (sec. 6.1.7) [44].

Desde el punto de vista de las funcionalidades, no se aprecian novedades relevantes respecto a las herramientas de e-learning analizadas en [39]. La única novedad destacable es la integración con redes sociales. Sin embargo, sí es posible observar una clara tendencia que las hace converger hacia modelos de explotación online. Aunque existen precedentes muy prestigiosos de herramientas comerciales de e-learning que operan bajo ese modelo [1], tal alternativa se ha convertido en la norma en las herramientas analizadas [6, 15, 26, 37, 61, 83, 121, 126]. De hecho, las herramientas de e-learning que siguen operando mediante el modelo de aplicaciones de escritorio, se han trasladado hacia un entorno de explotación a través de herramientas de conferencia de amplia expansión, por ejemplo *Skype* [107], sobre las que se instalan *Add-ons* con funcionalidades específicas [120]. Además, existen herramientas que combinan tanto el modelo online como la integración en herramientas de conferencia [78, 131, 132].

La principal ventaja de este tipo de herramientas web es que reducen notablemente los problemas de compatibilidad, puesto que son desarrollos multi-plataforma que se pueden usar desde diferentes navegadores con independencia tanto del hardware como del sistema operativo. Este hecho hace posible que sean accesibles también desde dispositivos móviles. Además, libera a los usuarios de las tareas de instalación, configuración y actualización del software de la plataforma, más allá del necesario para poder visualizar ciertos componentes web avanzados. Por último, reducen el consumo de recursos en los equipos de los usuarios, por ejemplo, el espacio ocupado en el disco duro. Un inconveniente es la disponibilidad, ya que un fallo en el servidor donde transcurre la actividad de e-learning paraliza la misma, si bien es posible mitigar tal inconveniente mediante la replicación de las actividades en varios servidores.

Por otro lado, al final de la década pasada el uso de plataformas sociales virtuales, como *Second Life* [104], ha surgido como alternativa a las herramientas de e-learning síncrono tradicionales, aunando tanto actividades educativas como relaciones sociales [48]. El propósito de este tipo de plataformas es fomentar la cooperación y la sensación de presencia y de pertenencia al grupo. En las plataformas sociales los usuarios interaccionan entre ellos a través de objetos virtuales personalizables denominados avatares. Un avatar representa el *alter ego* de un participante, a través del cual se desenvuelve dentro del mundo virtual de la plataforma, participa en clases y actividades, y dialoga con compañeros con los que comparte intereses similares, lo que promueve la colaboración y

aprendizaje mutuos. Concretamente, se han publicado trabajos científicos acerca del uso de plataformas sociales como herramientas de e-learning síncrono en varias universidades [12, 23, 71, 77], especialmente en ramas de la salud como medicina y enfermería [2, 3, 16, 52, 101, 112]. También han sido utilizadas como herramienta de capacitación en organizaciones [122] y empresas [72].

Existen múltiples beneficios del uso de plataformas sociales virtuales como herramientas de e-learning síncrono [12, 136]. La emulación de actividades reales mediante mundos virtuales en tres dimensiones estimula al usuario, lo que facilita su aprendizaje. Estos entornos virtuales resultan atractivos, e incluso eliminan barreras en los usuarios más retraídos, lo que incrementa la comunicación y la cooperación, por lo que el proceso de aprendizaje se vuelve más fluido. Además, debido a que estas plataformas virtuales tratan de asemejarse lo máximo posible al mundo real, promueve que los usuarios entrenen habilidades, o prueben nuevas ideas, que puedan llevarse a la práctica posteriormente. En esta misma línea, es posible emular entornos peligrosos de una manera segura y evitar riesgos relacionados tanto con el uso de sustancias peligrosas, como con la manipulación de material delicado de laboratorio. También permiten eliminar los problemas cotidianos de demanda, disponibilidad y reserva de equipamientos, laboratorios u otros recursos.

Capítulo 5

Conclusiones y trabajo futuro

Uno de los objetivos principales de la tesis era identificar, de manera precisa, la aplicación del paradigma de la Computación Autónoma (CA) en el ámbito de los sistemas de comunicación multimedia en tiempo real interactivos. La tesis realiza aportaciones interesantes en este sentido, ya que enuncia una serie de características autónomas que pueden ser observadas en los SCM en general, y en dichos sistemas en particular. Además, ha sido posible relacionar tales características con los fundamentos iniciales del paradigma de la CA, formulándolas como subpropiedades del conjunto inicial de propiedades autónomas, y también establecer relaciones entre ellas, dando lugar a la creación de un tejido de conocimiento que permite comprender mejor el propósito de tales subpropiedades, así como las colaboraciones y dependencias entre ellas. Un segundo aporte a destacar es la taxonomía propuesta para clasificar las diferentes técnicas que permiten implementar una determinada subpropiedad autónoma en un sistema de comunicación multimedia en tiempo real. Dicha taxonomía fue validada aplicándola sobre un conjunto destacado de publicaciones científicas, en términos de relevancia, relacionadas con las plataformas de distribución multimedia en tiempo real.

Del análisis realizado a dicho conjunto de plataformas se desprende que, debido a su complejidad, disponen de características autónomas para desarrollar tareas de auto-gestión. Concretamente, subpropiedades autónomas como la auto-organización, el auto-despliegue, la auto-diagnos y la auto-estabilización son comunes en ese tipo de plataformas. Otras subpropiedades autónomas, como la auto-regulación o la auto-distribución, tienen cada vez más presencia, especialmente en plataformas orientados a las comunicaciones a gran escala. Es de esperar que el crecimiento, expansión y mejora de las plataformas de distribución multimedia dependerá de la constante evolución, y mayor adopción, de la implementación de subpropiedades autónomas que faciliten su auto-gestión. Del estudio realizado se extrae también que las subpropiedades autónomas relacionadas con la auto-protección resultan las menos extendidas. No obstante, algunas de ellas resultan de gran interés, como la auto-contención y la auto-inmunidad, por lo que es de esperar que surjan líneas de investigación

de las que se obtengan nuevas funcionalidades y técnicas en ese sentido.

El trabajo futuro en esta parcela se centrará en el análisis de las nuevas técnicas, enfoques o funcionalidades que puedan surgir en relación a la implementación de alguna de las subpropiedades autonómicas dentro del ámbito de los sistemas de comunicación multimedia en tiempo real. De igual manera, la taxonomía presentada en esta tesis doctoral quizás deba ser adaptada para dar cabida a características novedosas que sean identificadas en las nuevas técnicas propuestas.

Otro de los objetivos fundamentales de la tesis doctoral era construir una plataforma de distribución multimedia que implementase las características autonómicas más relevantes extraídas del análisis del paradigma de la CA. A tal respecto, se ha diseñado una plataforma autonómica que da cabida a las subpropiedades autonómicas más extendidas en la literatura. Dicha plataforma ha sido evaluada exhaustivamente mediante simulación, y también se ha llevado a la práctica para dar soporte a actividades de formación a distancia en un escenario disperso real, como es el caso de la corporación ArcelorMittal. La plataforma dispone concretamente de las subpropiedades autonómicas de auto-despliegue, auto-organización, auto-diagnóstico, auto-estabilización y auto-regulación, las cuales ayudan a proporcionar un servicio eficiente de distribución de información multimedia en tiempo real.

El diseño de la plataforma está dividido en varias redes virtuales destinadas a labores específicas: distribución de la información multimedia, gestión de los participantes y control de la red de distribución multimedia; las dos últimas administradas de forma centralizada por el RP. El uso de protocolos estándares, como RTP para el transporte de la información multimedia o SIP para la señalización, permite que la plataforma sea interoperable con otras soluciones destinadas a la comunicación multimedia ya sean software o hardware. Además, el diseño modular a través de múltiples redes virtuales sin solapamiento garantiza la extensibilidad de la plataforma para, por ejemplo, añadir nuevas redes virtuales que permitan la administración de otros aspectos relativos a las actividades que se desarrollan sobre la plataforma como es el caso de la gestión de turnos de intervención. El principal aporte es el uso del protocolo SIP como mecanismo sobre el que construir las técnicas que dan lugar a la auto-curación de la plataforma, ya que representa una aplicación de dicho protocolo más allá de su propósito original.

La red virtual de reflectores, encargada de la distribución de la información multimedia, hace un uso eficiente de los recursos de red aprovechando la comunicación IP multicast si está disponible. Su topología de tipo *full-mesh* minimiza la latencia al garantizar que la información multimedia circula a través del menor número de elementos intermedios posible. Las experimentaciones llevadas a cabo para verificar el funcionamiento de la plataforma, donde se simulaban errores en los elementos intermedios de la red virtual de reflectores,

muestran que la plataforma es capaz de hacer frente a dichos errores sin que la calidad de la experiencia por parte del usuario se vea afectada. No obstante, la plataforma no elimina los riesgos e inconvenientes de depender de un elemento central. El RP gestiona tanto a los participantes de la actividad como la eficiencia de la organización de la red virtual de reflectores. Un fallo en el RP impide la unión de nuevos participantes a la actividad e imposibilita mantener la eficiencia de la red virtual de reflectores, al no poder ser adaptada en respuesta a los cambios que se producen durante el transcurso de la actividad. Sin embargo, como atenuante, un fallo en el RP no impide el desarrollo de las actividades en curso, aunque estas pueden verse degradadas en el tiempo en términos de eficiencia en las comunicaciones. Los problemas originados a raíz de la dependencia del RP pueden solventarse aplicando sobre el mismo soluciones tolerantes a fallos.

El trabajo futuro en esta línea va orientado, por una parte, a mejorar la fiabilidad del RP mediante soluciones software basadas en el protocolo SIP que permitan minimizar el riesgo de dependencia de un elemento central. Otra línea de actuación será implementar en la plataforma nuevas subpropiedades autonómicas como la auto-distribución, o ampliar otras ya existentes como la auto-regulación. Desde el punto de vista de la auto-distribución, se pretende trasladar el diseño de la plataforma hacia un modelo híbrido donde, en caso de sobrecarga en los reflectores, alguno de los participantes administrados por el reflector sobrecargado pueda ejercer también el rol de reflector y así balancear la carga de trabajo. Por otra parte, se pretende implementar la auto-regulación mediante técnicas de transcodificación de los datos en los reflectores. Así pues, será posible adaptar la calidad de la información multimedia distribuida en función de las condiciones de la red virtual de reflectores y de la calidad observada por los participantes. La aplicación de este tipo de técnicas de control de los medios retransmitidos puede dar lugar a que la plataforma disponga de técnicas de auto-destrucción. Esto ocurre si se detiene la retransmisión de un medio, como el vídeo, hacia o desde un participante para proteger la calidad de la experiencia de dicho participante o del conjunto en general.

Por último, la investigación llevada a cabo en el campo del e-learning síncrono muestra la tendencia, cada vez más amplia, de trasladar este tipo de plataformas hacia un modelo de explotación online, así como la integración con redes sociales. Por tanto, a corto plazo se deberá adaptar la herramienta e-pSyLon, junto con la plataforma de distribución multimedia, para adecuarlas de acuerdo a tales demandas. Desde el punto de vista del uso del e-learning síncrono, el caso de prueba en ArcelorMittal resultó exitoso y demuestra la creciente aceptación del uso de las herramientas de e-learning síncrono en procesos de formación, tanto en el ámbito corporativo, como académico. A este respecto, una línea de investigación futura será determinar el grado de beneficio sobre el aprendizaje del usuario final a raíz de utilizar la herramienta e-pSyLon.

Capítulo 6

Compendio de publicaciones

En este capítulo se adjuntan las publicaciones que conforman la tesis doctoral. Conviene destacar que se incluyen tanto los artículos publicados en medios con impacto, como son revistas indexadas en el JCR o similares, así como artículos presentados en congresos de menor relevancia. No obstante, estos últimos permiten ampliar la visión general de la labor investigadora llevada a cabo durante la realización de la tesis doctoral, por lo que su inclusión puede considerarse obligada para entender apropiadamente el alcance de la misma. Se incluyen también artículos surgidos a raíz de la investigación realizada durante la tesis doctoral, y que se encuentran actualmente en proceso de revisión.

6.1. Artículos aceptados

6.1.1. Self-* in Multimedia Communication Overlays

- P. Nuño, J. C. Granda, F. J. Suárez and D. F. García. Self-* in Multimedia Communication Overlays, *en Computer Communications (Elsevier)*, Volume 36, Issue 7, pp 817-833, April 2013. ISSN 0140-3664. DOI 10.1016/j.comcom.2012.12.009.

6.1.2. Automatic Deployment of a Communication Mesh for Synchronous e-Learning Activities

- P. Nuño, J. C. Granda, D. F. García, and F. J. Suárez. Automatic deployment of a communication mesh for synchronous e-learning activities, *en Proceedings of the 5th IARIA International Conference on Systems and Networks Communications (ICSNC 2010)*, Niza, France, Aug. 2010, pp. 213-220. DOI 10.1109/ICSNC.2010.40.

6.1.3. Autonomic Platform for Synchronous e-Training in Dispersed Organizations

- J. C. Granda, P. Nuño, D. F. García and F. J. Suárez. Autonomic platform for synchronous e-training in dispersed organizations, *en Journal of Network and Systems Management (Springer)*, pp 1-27, October 2013. DOI 10.1007/s10922-013-9290-4

6.1.4. Security Issues in a Synchronous e-Training Platform

- J. C. Granda, P. Nuño, D. F. García, and F. J. Suárez. Security issues in a synchronous e-training platform, *en Proceedings of the 6th International Conference on Availability, Reliability and Security (ARES'11)*, Vienna, Austria, Aug. 2011, pp. 485-492.
DOI 10.1109/ARES.2011.75.

6.1.5. Towards Resilient Synchronous e-Training Platforms

- J. C. Granda, P. Nuño, D. F. García, and F. J. Suárez. Towards resilient synchronous e-training platforms, *en Proceedings of the 10th IEEE International Symposium on Network Computing and Applications (NCA'11)*, Cambridge, MA, USA, Aug. 2011, pp. 251-256. DOI 10.1109/NCA.2011.42.

6.1.6. E-pSyLon: A Synchronous e-Learning Platform for Staff Training in Large Corporations

- J. C. Granda, P. Nuño, F. J. Suárez, and M. A. Pérez. E-pSyLon: A synchronous e-learning platform for staff training in large corporations, *en Multimedia Tools Applications (Springer)*, pp. 1-33, March 2012. DOI 10.1007/s11042-012-1061-9.

6.1.7. Synchronous E-Learning Tools

- J. C. Granda, P. Nuño, F. J. Suárez, and D. F. García. Chapter 6: Synchronous E-Learning Tools, *en E-Learning: New Technology, Applications and Future Trends*, ISBN 978-1-62618-961-4.

Este capítulo de libro se encuentra actualmente aceptado y pendiente de publicación. Por tanto, se adjunta también la carta del editor del libro a través de la cual se comunicó dicha aceptación.

6.2. Artículos en proceso de revisión

6.2.1. Resilient Overlay Network for Real-Time Interactive Multimedia Communications in Corporate Networks

- P. Nuño, J. C. Granda, J. Molleda, R. Usamentiaga and D. F. García. Resilient Overlay Network for Real-Time Interactive Multimedia Communications in Corporate Networks, *en Journal of Network and Computer Applications (Elsevier), en proceso de revisión.*

Capítulo 7

Informe sobre las publicaciones

En este capítulo se detalla la información referente a la calidad y el factor de impacto de las publicaciones que componen la tesis doctoral.

- 1 P. Nuño, J. C. Granda, F. J. Suárez and D. F. García. Self-* in multimedia communication overlays, *en Computer Communications (Elsevier)*, Volume 36, Issue 7, pp 817-833, April 2013. ISSN 0140-3664. DOI 10.1016/j.comcom.2012.12.009.

- Estado: Publicado online
- Factor de impacto (JCR 2012): 1.079
- Factor de impacto (5 años): 1.227
- Categorías:
 - COMPUTER SCIENCE, INFORMATION SYSTEMS 58/132 Q2:T2
 - ENGINEERING, ELECTRICAL & ELECTRONIC 126/242 Q3:T2
 - TELECOMMUNICATIONS 35/77 Q2:T2

- 2 P. Nuño, J. C. Granda, D. F. García, and F. J. Suárez. Automatic deployment of a communication mesh for synchronous e-learning activities, *en Proceedings of the 5th IARIA International Conference on Systems and Networks Communications (ICSNC 2010)*, Niza, France, Aug. 2010, pp. 213-220. DOI 10.1109/ICSNC.2010.40.

- Estado: Publicado
- Ratio Citas/Publicaciones del congreso: 121/68 (1.78)¹
- Tasa de aceptación: 30 %
- FAU²: A

¹Fuente: Microsoft Academic.

²Ranking de conferencias elaborado por la Universidad de Erlangen-Núremberg (Friedrich-Alexander-Universität Erlangen-Nürnberg).

- 3** J. C. Granda, P. Nuño, D. F. García and F. J. Suárez. Autonomic platform for synchronous e-training in dispersed organizations, *en Journal of Network and Systems Management (Springer)*, pp 1-27, October 2013. DOI 10.1007/s10922-013-9290-4
- Estado: Publicado online
 - Factor de impacto (JCR 2012): 0.432
 - Factor de impacto (5 años): 0.8
 - Categorías:
 - COMPUTER SCIENCE, INFORMATION SYSTEMS 110/132 Q4:T3
 - TELECOMMUNICATIONS 64/77 Q4:T3
- 4** J. C. Granda, P. Nuño, D. F. García, and F. J. Suárez. Security issues in a synchronous e-training platform, *en Proceedings of the 6th International Conference on Availability, Reliability and Security (ARES'11)*, Vienna, Austria, Aug. 2011, pp. 485-492. DOI 10.1109/ARES.2011.75.
- Estado: Publicado
 - Ratio Citas/Publicaciones del congreso: 267/114 (2.34)
 - Tasa de aceptación: <25 %, n/a en workshop
 - ERA 2010: B
 - CORE 2008: B
 - FAU: B
- 5** J. C. Granda, P. Nuño, D. F. García, and F. J. Suárez. Towards resilient synchronous e-training platforms, *en Proceedings of the 10th IEEE International Symposium on Network Computing and Applications (NCA'11)*, Cambridge, MA, USA, Aug. 2011, pp. 251-256. DOI 10.1109/NCA.2011.42.
- Estado: Publicado
 - Ratio Citas/Publicaciones del congreso: 127/57 (2.23)
 - Tasa de aceptación: 25 %, 60 % en workshop
 - ERA 2010: A
 - CORE 2008: A
 - FAU: A

6 J. C. Granda, P. Nuño, F. J. Suárez, and M. A. Pérez. E-pSyLon: A synchronous e-learning platform for staff training in large corporations, *en Multimedia Tools Applications (Springer)*, pp. 1-33, March 2012. DOI 10.1007/s11042-012-1061-9.

- Estado: Publicado online
- Factor de impacto (JCR 2012): 1.014
- Factor de impacto (5 años): 0.932
- Categorías:
 - COMPUTER SCIENCE, INFORMATION SYSTEMS 63/132 Q2:T2
 - COMPUTER SCIENCE, SOFTWARE ENGINEERING 50/105 Q2:T2
 - COMPUTER SCIENCE, THEORY & METHODS 33/100 Q2:T1
 - ENGINEERING, ELECTRICAL & ELECTRONIC 130/242 Q2:T2

7 J. C. Granda, P. Nuño, F. J. Suárez, and D. F. García. Chapter 6: Synchronous e-learning tools, *en E-Learning: New Technology, Applications and Future Trends*, ISBN 978-1-62618-961-4.

- Estado: Publicado
- Fecha de publicación: 19 de octubre de 2013
- Editorial: Nova Publishers
- Serie: Education in a Competitive and Globalizing World
- Serie incluida en el *Book Citation Index* del Web of Knowledge (Thomson Reuters, 2013)
- Ranking Editorial: 17^a de 21 totales de acuerdo al *Cambridge Scientific Abstracts* (2011)

8 P. Nuño, J. C. Granda, J. Molleda, R. Usamentiaga and D. F. García. Resilient Overlay Network for Real-Time Interactive Multimedia Communications in Corporate Networks, *en Journal of Network and Computer Applications (Elsevier)*.

- Estado: En proceso de revisión
- Factor de impacto (JCR 2012): 1.467

- Factor de impacto (5 años): 1.251
- Categorías:
 - COMPUTER SCIENCE, HARDWARE & ARCHITECTURE
11/50 Q1:T1
 - COMPUTER SCIENCE, INTERDISCIPLINARY APPLICA-
TIONS 44/90 Q2:T2
 - COMPUTER SCIENCE, SOFTWARE ENGINEERING 24/105
Q1:T1

Bibliografía

- [1] Adobe. Accesible en:
<http://www.adobe.com/es/products/adobecconnect.html>.
- [2] M. Aebersold, D. Tschannen, M. Stephens, P. Anderson, and X. Lei. Second Life: a new strategy in educating nursing students. *Clinical Simulation in Nursing*, pages 1–7, 2011.
- [3] N. Ahern and D. M. Wink. Virtual learning environments: Second life. *Nurse Educator*, 35(6):225–227, 2010.
- [4] L. O. Alima, S. Haridi, A. Ghodsi, S. El-Ansary, and P. Brand. Self-properties in distributed k-ary structured overlay networks. In *Proc. 1st Int. Workshop on Self-* Properties in Complex Information Systems (SELF-* 2004)*, Bertinoro, Forlì-Cesena, Italy, June 2004.
- [5] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, and J. Linden. Internet Low Bit Rate Codec (iLBC). RFC 3951 (Experimental), December 2004.
- [6] AnyMeeting. Accesible en: <http://www.anymeeting.com/>.
- [7] P. Baccichet, J. Noh, E. Setton, and B. Girod. Content-aware P2P video streaming with low latency. In *Proc. 8th IEEE Int. Conf. on Multimedia and Expo (ICME 2007)*, pages 400–403, Beijing, China, July 2007.
- [8] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. 17th ACM Int. Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2002)*, pages 205–217, Pittsburgh, PA, USA, August 2002.
- [9] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller. OMNI: An efficient overlay multicast infrastructure for real-time applications. *Computer Networks*, 50(6):826–841, 2006.
- [10] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP). RFC 3711, IETF, 2004.

- [11] L. Beaudoin, N. Japkowicz, and S. Matwin. Autonomic computer network defence using risk state and reinforcement learning. In *Proc. 1st Int. Conf. on Cyber Warfare (CCD COE)*, Tallinn, Estonia, June 2009.
- [12] L. M. Beltrán Sierra, R. S. Gutiérrez, and C. L. Garzón-Castro. Second life as a support element for learning electronic related subjects: A real case. *Computers & Education*, 58(1):291–302, 2012.
- [13] A. Berns and S. Ghosh. Dissecting Self-* properties. In *Proc. 3rd IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems (SASO 2009)*, volume 0, pages 10–19, Los Alamitos, CA, USA, August 2009.
- [14] A. Bikfalvi, J. García-Reinoso, I. Vidal, and F. Valera. A peer-to-peer IPTV service architecture for the IP multimedia subsystem. *International Journal of Communication Systems*, 23(6-7):780–801, 2010.
- [15] Blackboard Collaborate. Accesible en:
<https://www.blackboard.com/platforms/collaborate/overview.aspx>.
- [16] M. N. K. Boulos, L. Hetherington, and S. Wheeler. Second life: an overview of the potential of 3-d virtual worlds in medical and health education. *Health Information & Libraries Journal*, 24(4):233–245, 2007.
- [17] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml) 1.0 (fifth edition). World Wide Web Consortium, Recommendation REC-xml-20081126, 2008.
- [18] W. B. Cannon. Organization for physiological homeostasis. *Physiological Reviews*, 9:399–431, 1929.
- [19] M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: High-bandwidth multicast in cooperative environments. *ACM Operating Systems Review*, 37(5):298–313, 2003.
- [20] Y. Chawathe, S. McCanne, and E. A. Brewer. RMX: Reliable multicast for heterogeneous networks. In *Proc 19th IEEE Int. Conf. on Computer Communications (INFOCOM 2000)*, volume 2, pages 795–804, Tel-Aviv, Israel, March 2000.
- [21] F. Chiang, J. Agbinya, and R. Braun. Risk and vulnerability assessment of secure autonomic communication networks. In *Proc. 2nd Int. Conf. on Wireless Broadband and Ultra Wideband Communications (AUSWIRELESS '07)*, pages 40–46, Washington, DC, USA, April 2007.

-
- [22] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang. A case for end system multicast. *IEEE Journal on Selected Areas in Communications*, 20(8):1456–1471, 2002.
- [23] L. Y. Chung. Virtual reality in college english curriculum: Case study of integrating second life in freshman english course. In *Proc IEEE 26th Int. Conf. Advanced Information Networking and Applications Workshops, WAINA 2012*, volume 0, pages 250–253, Los Alamitos, CA, USA, 2012.
- [24] E. Damiani, D. C. Di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proc. 9th ACM Int. Conf. on Computer and Communications Security (CCS '02)*, pages 207–216, New York, NY, USA, November 2002.
- [25] T. De Wolf and T. Holvoet. A taxonomy for self-properties in decentralised autonomic computing. *Autonomic Computing: Concepts, Infrastructure, and Applications*, pages 101–120, 2006.
- [26] DimDim. Accesible en: <http://www.dimdim.com>.
- [27] Y. Ding, J. Liu, D. Wang, and H. Jiang. Peer-to-peer video-on-demand with scalable video coding. *Computer Communications*, 33(14):1589–1597, 2010.
- [28] C. Diot, W. Dabbous, and J. Crowcroft. Multipoint communication: A survey of protocols, functions, and mechanisms. *IEEE Journal on Selected Areas in Communications*, 15(3):277–290, 1997.
- [29] C. Diot, B. Neil, L. Bryan, and K. D. Balensiefen. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 14(1):78–88, 2000.
- [30] S. Dobson, R. Sterritt, P. Nixon, and M. Hinchey. Fulfilling the vision of autonomic computing. *Computer*, 43(1):35–41, 2010.
- [31] S. Dolev. *Self-stabilization*. MIT Press, Cambridge, MA, USA, 2000.
- [32] A. El-Sayed, V. Roca, and L. Mathy. A survey of proposals for an alternative group communication service. *IEEE Network*, 17(1):46–51, 2003.
- [33] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. RFC 2117, IETF, 1997. Obsoleted by RFC 2362.

- [34] P. A. Felber and E. W. Biersack. Self-scaling networks for content distribution. In *Proc. 1st Int. Workshop on Self-* Properties in Complex Information Systems (SELF-* 2004)*, Bertinoro, Forlì-Cesena, Italy, June 2004.
- [35] P. A. Felber and E.W. Biersack. Cooperative content distribution: Scalability through self-organization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 3460 of LNCS:343–357, 2005.
- [36] P. Flocchini, G. Prencipe, and N. Santoro. Self-deployment of mobile sensors on a ring. *Theoretical Computer Science*, 402 (1):67–80, 2008.
- [37] Fuze Box. Accesible en: <https://www.fuzebox.com/>.
- [38] A. G. Ganek and T. A. Corbi. The dawning of the autonomic computing era. *IBM Systems Journal*, 42(1):5–18, 2003.
- [39] J. C. Granda. *Caracterización, evaluación y optimización de sistemas multimedia interactivos en entornos de e-learning síncrono*. PhD thesis, Universidad de Oviedo, July 2008.
- [40] J. C. Granda, D. F. García, P. Nuño, and F. J. Suárez. An efficient networking technique for synchronous e-learning platforms in corporate environments. *Computer Communications*, 33(14):1752–1766, 2010.
- [41] J. C. Granda, P. Nuño, D. F. García, and F. J. Suárez. Security issues in a synchronous e-training platform. In *Proc. of the IEEE 6th Int. Conf. on Availability, Reliability and Security (ARES'11)*, pages 485–492, Vienna, Austria, August 2011.
- [42] J. C. Granda, P. Nuño, D. F. García, and F. J. Suárez. Towards resilient synchronous e-training platforms. In *Proc. of the IEEE 10th Int. Symp. on Network Computing and Applications (NCA '11)*, pages 251–256, Cambridge, MA, USA, August 2011.
- [43] J. C. Granda, P. Nuño, D. F. García, and F. J. Suárez. Autonomic platform for synchronous e-training in dispersed organizations. *Journal of Network and Systems Management*, pages 1–27, 2013.
- [44] J. C. Granda, P. Nuño, F. J. Suárez, and D. F. García. *E-Learning: New Technology, Applications and Future Trends. Chapter 6: Synchronous E-Learning Tools*. Nova Science Pub Incorporated, 2013.

-
- [45] J. C. Granda, P. Nuño, F. J. Suárez, and M. A. Pérez. E-pSyLon: A synchronous e-learning platform for staff training in large corporations. *Multimedia Tools and Applications*, 66(3):431–463, 2013.
- [46] D. Hales and S. Arteconi. SLACER: A self-organizing protocol for coordination in peer-to-peer networks. *IEEE Intelligent Systems*, 21(2):29–35, 2006.
- [47] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327, IETF, 1998.
- [48] K. Haycock and J. W. Kemp. Immersive Learning Environments in Parallel Universes: Learning through Second Life. *School Libraries Worldwide*, 14(2):89–97, 2008.
- [49] A. R. Haydarlou, B. J. Overeinder, M. A. Oey, and F. M. T. Brazier. Multi-level model-based self-diagnosis of distributed object-oriented systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4158 of LNCS:67–77, 2006.
- [50] M. M. Hefeeda, B. K. Bhargava, and D. K. Y. Yau. A hybrid architecture for cost-effective on-demand media streaming. *Computer Networks*, 44(3):353–382, 2004.
- [51] M. M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. K. Bhargava. PROMISE: Peer-to-peer media streaming using collectcast. In *Proc. 11th ACM Int. Conf. on Multimedia (MM '03)*, pages 45–54, Berkeley, CA, USA, November 2003.
- [52] M. Hermanns and C. Kilmon. Second Life as a clinical conference environment: Experience of students and faculty. *Clinical Simulation in Nursing*, 08:297–300, 2011.
- [53] E. Hladká, P. Holub, and J. Denemark. Teleconferencing support for small groups. In *Proc. TERENA Networking Conf.*, pages 1–7, Limerick, Ireland, June 2002.
- [54] E. Hladká, P. Holub, and J. Denemark. User-empowered programmable network support for collaborative environment. In *Proc. of the 3rd Euro. Conf. on Universal Multiservice (ECUMN 2004)*, volume 3262 of LNCS, pages 367–376, Porto, Portugal, October 2004.
- [55] P. Horn. Autonomic computing: IBM's Perspective on the State of Information Technology, 2001.

- [56] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas. A survey of Application-Layer Multicast protocols. *IEEE Communications Surveys & Tutorials*, 9:58–74, 2007.
- [57] A. Howard, M. J. Matarić, and S. S. Gaurav. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots*, 13:113–126, 2002.
- [58] N. Huang, Y. Tzang, H. Chang, and C. Ho. Enhancing P2P overlay network architecture for live multimedia streaming. *Information Sciences*, 180(17):3210–3231, 2010.
- [59] M. C. Huebscher and J. A. McCann. A survey of autonomic computing: degrees, models, and applications. *ACM Computing Surveys*, 40(3):7:1–7:28, 2008.
- [60] K. Hyder, A. Kwinn, R. Miazga, and M. Murray. *The eLearning Guild's handbook on synchronous e-Learning*. The eLearning Guild, 2007.
- [61] iLinc. Accesible en: www.ilinc.com/.
- [62] B. Jacob, R. Lanyon-Hogg, D. K. Nadgir, and A. F. Yassin. *A Practical Guide to the IBM Autonomic Computing Toolkit*. IBM Redbooks. Redbooks, IBM, 2004.
- [63] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr. Overcast: Reliable multicasting with on overlay network. In *Proc. 4th USENIX Int. Conf. and Symp. on Operating System Design & Implementation (OSDI '00)*, pages 14–29, San Diego, CA, USA, October 2000.
- [64] H. Kalva and J. B. Lee. The VC-1 video coding standard. *IEEE MultiMedia*, 14(4):88–91, 2007.
- [65] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36:41–50, 2003.
- [66] V. Kompella, J. Pasquale, and G. Polyzos. Two distributed algorithms for multicasting multimedia information. In *Proc. IEEE Int. Conf. on Communications (ICC '93)*, pages 343–349, Geneva, Italy, May 1993.
- [67] L. Lambrinos and E. Demetriou. An adaptive live media streaming architecture. In *Proc. 2nd IARIA Int. Conf. on Advances in Multimedia (MMEDIA '10)*, pages 74–77, Athens/Glyfada, Greece, June 2010.

-
- [68] B. Li and K. Nahrstedt. A control-based middleware framework for quality of service adaptations. *IEEE Journal of Selected Areas in Communication, Special Issue on Service Enabling Platforms*, 17(9):1632–1650, 1999.
- [69] Z. Liu, G. He, and Z. Liu. Metrino: An application-level multicast system for real-time video communication. In *Proc. Int. Conf. on Wireless Communications, Networking and Mobile Computing (WCNM 2005)*, pages 1265–1270, Wuham, China, September 2005.
- [70] Z. Liu, Y. Shen, K. W. Ross, S. S. Panwar, and Y. Wang. LayerP2P: using layered video chunks in P2P live streaming. *IEEE Transactions on Multimedia*, 11(7):1340–1352, 2009.
- [71] A. De Lucia, R. Francese, I. Passero, and G. Tortora. Development and evaluation of a virtual campus on Second Life: The case of SecondDMI. *Computers & Education*, 52(1):220–233, 2009.
- [72] P. Ludlow and M. Wallace. *The Second Life Herald: The Virtual Tabloid that Witnessed the Dawn of the Metaverse*. The MIT Press, 2007.
- [73] C. Luo, J. Li, and S. Li. DigiMetro: An application-level multicast system for multi-party video conferencing. In *Proc. 23rd IEEE Int. Conf. on Global Telecommunications (GLOBECOM '04)*, volume 2, pages 982–987, Dallas, Texas, USA, November-December 2004.
- [74] M. Mamei, R. Menezes, R. Tolksdorf, and F. Zambonelli. Case studies for self-organization in computer science. *Journal of Systems Architecture*, 52:443–460, 2006.
- [75] M. Mao, J. Li, and M. Humphrey. Cloud auto-scaling with deadline and budget constraints. In *Proc. 11th IEEE/ACM Int. Conf. on Grid Computing (GRID 2010)*, pages 41–48, Brussels, Belgium, October 2010.
- [76] G. Maraviglia, M. Masi, V. Merlo, F. Licandro, A. Russo, and G. Schembra. Synchronous multipoint e-Learning realized on an intelligent software-router platform over unicast networks: Design and performance issues. In *Proc. 12th IEEE Int. Conf on Emerging Technologies and Factory Automation (ETFA 2007)*, pages 1172–1179, Patras, Greece, September 2007.
- [77] M. Mayrath, J. Sanchez, T. Traphagan, J. Heikes, and A. Trivedi. Using Second Life in an English Course: Designing Class Activities to Address Learning Objectives. In *Proc. 15th ACE World Conference on Educational Multimedia, Hypermedia&Telecommunications (EdMEDIA '07)*, Vancouver, Canada, June 2007.

- [78] Mikogo. Accesible en: <http://www.mikogo.com/>.
- [79] H. A. Müller, L. O'Brien, M. Klein, and B. Wood. Autonomic computing. Technical report, Software Engineering Institute, Carnegie Mellon University, 2006.
- [80] J. J. D. Mol, J. A. Pouwelse, M. Meulpolder, D. H. J. Epema, and H. J. Sips. Give-to-get: Free-riding-resilient video-on-demand in P2P systems. In *Proc. SPIE Int. Conf. on Multimedia Computing and Networking Conference (MMCN 2008)*, volume 6818, San Jose, CA, USA, January 2008.
- [81] J. Molleda, R. Usamentiaga, D. F. García, and F. G. Bulnes. Towards autonomic computing in machine vision applications. Techniques and strategies for a 3D reconstruction system in harsh industrial environments. In *Proc. SPIE on Image Processing: Machine Vision Applications IV*, volume 7877, pages 1–13, San Francisco, CA, USA, January 2011.
- [82] X. Navarro. Physiology of the autonomic nervous system. *Revista de Neurología*, 35(6):553–562, 2002.
- [83] Netviewer. Accesible en: <http://www.netviewer.com>.
- [84] NS-3. Accesible en: <http://www.nsnam.org/>.
- [85] P. Nuño, J. C. Granda, D. F. García, and F. J. Suárez. Automatic deployment of a communication mesh for synchronous e-learning activities. In *Proc. 5th IARIA Int. Conf. on Systems and Networks Communications, ICSNC '10*, pages 213–220, Niza, France, August 2010.
- [86] P. Nuño, J. C. Granda, J. Molleda, R. Usamentiaga, and D. F. García. Resilient overlay network for real-time interactive multimedia communications in corporate networks. *Journal of Network and Computer Applications*, En proceso de revisión., 2013.
- [87] P. Nuño, J. C. Granda, F. J. Suárez, and D. F. García. Self-* in multimedia communication overlays. *Computer Communications*, 36(7):817–833, April 2013.
- [88] M. J. Oudshoorn, M. M. Fuad, and D. Deb. Towards autonomic computing: Injecting self-organizing and self-healing properties into java programs. In *Proc. 5th Int. Conf. on New Trends in Software Methodologies, Tools and Techniques (SOMET 06)*, pages 384–406, Québec, Canada, October 2006.

-
- [89] M. Parashar and S. Hariri. Autonomic computing: An overview. In *Unconventional Programming Paradigms*, pages 247–259. Springer Verlag, 2005.
- [90] C. Z. Patrikakis, Y. Despotopoulos, A. M. Robotis, and A. L. Lambiris. PERIPHLEX: multicast delivery using core unicast distribution with peripheral multicast reflectors. In *Proc. 12th Int. Conf. on World Wide Web (WWW2003)*, page 292, Budapest, Hungary, May 2003.
- [91] A. H. Payberah, J. Dowling, F. Rahimian, and S. Haridi. GradientTv: Market-based P2P live media streaming on the gradient overlay. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6115:212–225, 2010.
- [92] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An application level multicast infrastructure. In *Proc. 3rd USENIX Symp. on Internet Technologies and Systems (USITS 2001)*, pages 49–60, San Francisco, CA, USA, March 2001.
- [93] C. Perkins. *RTP: audio and video for the internet*. Addison-Wesley Professional, first edition, 2003.
- [94] F. Pianese, D. Perino, J. Keller, and E. W. Biersack. PULSE: An adaptive, incentive-based, unstructured P2P live streaming system. *IEEE Transactions on Multimedia*, 9(8):1645–1660, 2007.
- [95] J. Postel. Transmission Control Protocol. RFC 793, IETF, 1981.
- [96] X. Qiu, W. Huang, C. Wu, F. C. M. Lau, and X. Lin. InstantLeap: An architecture for fast neighbor discovery in large-scale P2P VoD streaming. *Multimedia Systems*, 16(3):183–198, 2010.
- [97] M. Rahman, R. Ranjan, and R. Buyya. A taxonomy of autonomic application management in grids. In *Proc. 16th IEEE Int. Conf. on Parallel and Distributed Systems (ICPADS 2010)*, pages 189–196, Shanghai, China, June 2010.
- [98] V. Roca and A. El-Sayed. Host-based multicast (HBM) solution for group communications. In *Proc. 1st IEEE Int. Conf. on networking (ICN 2001)*, volume 2093, pages 610–619, Colmar, France, July 2001.
- [99] J. Rosenberg. A framework for conferencing with the Session Initiation Protocol (SIP). RFC 4353, IETF, 2006.

- [100] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, IETF, 2002.
- [101] B. Schmidt and S. Stewart. Implementing the virtual reality learning environment: Second Life. *Nurse Educator*, 34(4):152–155, 2009.
- [102] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, IETF, 2003.
- [103] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326, IETF, 1998.
- [104] Second Life. Información sobre la plataforma virtual en: <http://secondlife.com/>.
- [105] M. W. Shapiro. Self-healing in modern operating systems. *Queue*, 2:66–75, 2004.
- [106] F. Silla and J. Duato. High-performance routing in networks of workstations with irregular topology. *IEEE Transactions on Parallel and Distributed Systems*, 11(7):699–719, 2000.
- [107] Skype. Accesible en: <http://www.skype.com/>.
- [108] H. Song, D. S. Lee, and H. R. Oh. Application layer multicast tree for real-time media delivery. *Computer Communications*, 29(9):1480–1491, 2006.
- [109] R. Sterritt. Autonomic computing. *Innovations in Systems and Software Engineering*, 1(1):79–88, 2005.
- [110] R. Sterritt and M. Hinchey. SPAACE IV: Self-properties for an autonomous & autonomic computing environment - Part IV A Newish hope. In *Proc. 7th IEEE Int. Conf. and Workshops on Engineering of Autonomic and Autonomous Systems (EASe '10)*, pages 119–125, Oxford, England, March 2010.
- [111] R. Sterritt, M. Parashar, H. Tianfield, and R. Unland. A concise introduction to autonomic computing. *Advanced Engineering Informatics*, 19(3):181–187, 2005.
- [112] S. Stewart, D. Pope, and D. Duncan. Using second life to enhance accel an online accelerated nursing bsn program. *Studies in Health Technology and Informatics*, 146:636–640, 2009.

-
- [113] H. Tianfield and R. Unland. Towards autonomic computing systems. *Engineering Applications of Artificial Intelligence*, 17(7):689–699, 2004.
- [114] D. A. Tran, K. Hua, and T. Do. A peer-to-peer architecture for media streaming. *IEEE Journal on Selected Areas in Communications*, 22(1):121–133, 2004.
- [115] R. Ubell. Engineers turn to e-learning. *IEEE Spectrum*, 37(10):59–63, 2000.
- [116] T. Čičić and H. Bryhni. Multicast-unicast reflector. In *Proc. Norsk Informatikk Konferanse (NIK'1998)*, pages 103–114, Kristiansand, Norway, November 1998.
- [117] T. Čičić, H. Bryhni, and S. Sølje. Unicast extensions to IP multicast. In *Proc. 5th Int. Conf. on Protocols for Multimedia Systems (PROMS 2000)*, pages 60–69, Krakow, Poland, October 2000.
- [118] V. Venkataraman, K. Yoshida, and P. Francis. Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast. In *Proc. 14th IEEE Int. Conf. on Network Protocols (ICNP 2006)*, pages 2–11, Santa Barbara, CA, USA, November 2006.
- [119] E. Vieira, G. De Sousa, H. Neves, A. V. Alves, E. A. Gomes, G. H. Matos, G. Lemos, and T. Aires. A strategy of multimedia reflectors to encryption and codification in real time. In *Proc. 8th IEEE Int. Symp. on Multimedia (ISM 2012)*, pages 278–285, Irvine, CA, USA, December 2012.
- [120] Vivu. Accesible en: <http://vivu.tv/vivuwweb/support/vuroom/>.
- [121] Vyew. Accesible en: <http://vyew.com/s/>.
- [122] C. X. Wang, H. Song, D. E. Stone, and Q. Yan. Integrating second life into an EFL program in china: Research collaboration across the continents. *TechTrends*, 53(6):14–19, 2009.
- [123] F. Wang, Y. Xiong, and J. Liu. MTreebone: A collaborative tree-mesh overlay network for multicast video streaming. *IEEE Transactions on Parallel and Distributed Systems*, 21(3):379–392, 2010.
- [124] W. Wang and Y. Chen. SmartPeerCast: A smart QoS driven P2P live streaming framework. *Multimedia Tools and Applications*, 54(2):445–471, 2011.
- [125] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.

- [126] WizIQ. Accesible en: <http://www.wiziq.com/>.
- [127] I. Wladawsky-Berger. Advancing e-business into the future: The grid. In *IBM Research*, New York, November 29 2001.
- [128] L. Yamamoto and G. Leduc. Autonomous reflectors over active networks. In *Symp. on Software Mobility and Adaptive Behaviour (AISB'01)*, pages 125–146, York, UK, March 2001.
- [129] C. K. Yeo, B. S. Lee, and M. H. Er. A survey of application level multicast techniques. *Computer Communications*, 27(15):1547–1568, 2004.
- [130] H. Yin, C. Lin, Q. Zhang, Z. Chen, and D. Wu. TrustStream: A secure and scalable architecture for large-scale internet media streaming. *IEEE Trans on Circuits and Systems for Video Technology*, 18(12):1692–1702, 2008.
- [131] Yugma. Accesible en: <https://www.yugma.com/>.
- [132] Yuuguu. Accesible en: <https://www.yuuguu.com/>.
- [133] X. Zhang, C. Li, and X. Li. Multi-party videoconferencing based on hybrid multicast with peer-forwarding. In *Proc. 16th IEEE Int. Conf. on Parallel and Distributed Systems (ICPADS)*, pages 812–816, Shanghai, China, December 2010.
- [134] X. Zhang and X. Li. Native multicast based videoconference service on CERNET. In *Proc Int Conf. on Communication Technology (ICCT'06)*, pages 1–4, Guilin, China, November 2006.
- [135] X. Zhang, J. Liu, B. Li, and T. P. Yum. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming. In *Proc 24th IEEE Int. Conf. on Computer Communications (INFOCOM 2005)*, volume 3, pages 2102–2111, Miami, FL, USA, March 2005.
- [136] Y. Zhao and L. Wu. Second life: A new window for e-learning. In *Proceedings of IEEE 9th Int. Conf. on Hybrid Intelligent Systems (HIS '09)*, volume 3, pages 191–194, Washington, DC, USA, August 2009.
- [137] J. Zhu, T. Tsuchiya, and K. Koyanagi. Peer-to-peer collaborative application-level multicast. In *Proc. 9th IEEE Int. Conf. on E-Commerce Technology; 4th IEEE Int. Conf. on Enterprise Computing, E-Commerce and E-Services (CEC/EEE 2007)*, pages 228–235, Tokyo, Japan, July 2007.