UNIVERSIDAD DE OVIEDO

Departamento de Informática

Tesis

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

Directores: Francisco José Suárez Alonso
Juan Carlos Granda Candás

Gijón, Octubre 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 Chencho. 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, no creo que esta labor 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 Autonómica en el ámbito de la distribución de contenidos multimedia en tiempo real interactivos. El germen de la computación autonómica 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 autonómicos tratando de identificar aquellas que pueden estar presentes en un sistema de comunicación multimedia. Posteriormente se han destacado las propiedades autonómicas 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 autonómicas 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

Índice de tablas

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 enseñaza 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 Autonómica** (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 autonómica 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 autonómico 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 autonómicos: 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 autonómicos, 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 autonómica. A continuación, se analizan las técnicas que permiten realizar comunicaciones multimedia de manera eficiente entre grupos de participantes dispersos.

3.1. Sistemas autonómicos

> *"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.*
>
> *No es un obstáculo relacionado con la 'inteligencia' de los computadores lo que amenaza nuestro progreso. Tiene poco que ver con desarrollar computadores 'inteligentes', que engloben el concepto popular de la Inteligencia Artificial para automatizar las operaciones diarias de un sistema informático. Puede resultar extraño viniendo de los creadores de Deep Blue, pero realmente no es necesario obtener un supercomputador que juege mejor al ajedrez, o máquinas sensibles y androides programados que amen o rían, para así superar el gran obstáculo que tenemos en el camino.*
>
> *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 Autonómica (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 dificultosa. 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 acuciante 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 autonómicos 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 autonómico

La esencia de los sistemas de computación autonómica 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 24×7). Los sistemas autonómicos 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 autonómicos, 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

[1] Característica 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].

un estado de equilibrio 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 autonómicos 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 autonómico existen una serie de propiedades o criterios fundamentales que un sistema debe satisfacer para ser considerado como tal [55]:

1. **Para poder considerarse autonómico, 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 autonómico.

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 autonómicos evitan a los usuarios un proceso de aprendizaje excesivo sobre cómo utilizarlos, interaccionar con ellos o interpretar sus resultados.

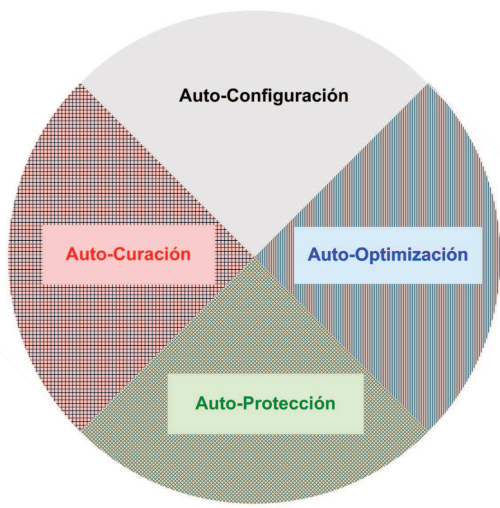


Figura 3.1: Propiedades fundamentales de los sistemas autonómicos.

3.1.2. Propiedades de los sistemas autonómicos

El propósito principal de los sistemas autonómicos es la auto-gestión (self-management). Para cumplir dicho propósito, y satisfacer las características enunciadas en el apartado anterior, los sistemas autonómicos 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 autonómicos.

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

Uno de los fundamentos principales de los sistemas autonómicos 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 autonómicos 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 autónomica, 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 autonómicos 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 autonómicos 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].

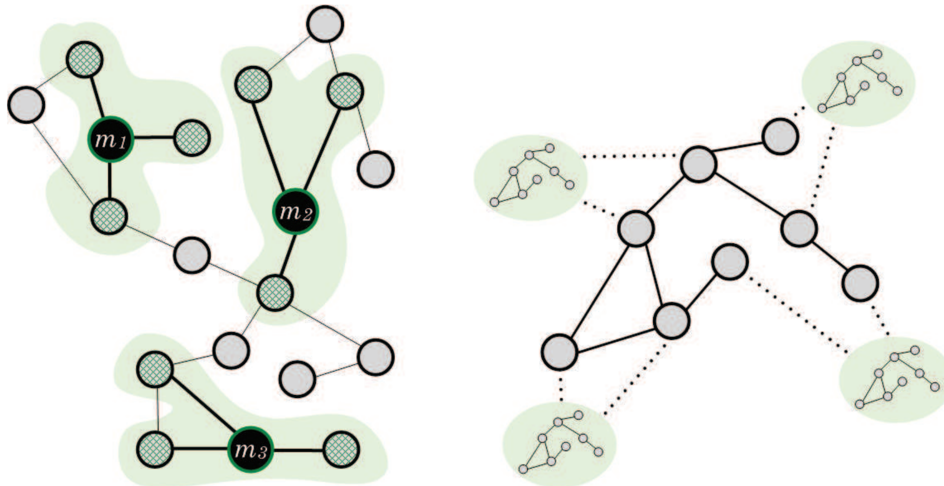
Las cuestiones anteriores han desembocado en que, por lo general, los proveedores de servicio de Internet no habilitan la posibilidad de comunicarse mediante IP multicast en sus redes, por lo que no es posible llevar a la práctica dicho modelo de comunicación para comunicar grupos geográficamente dispersos. 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.



(a) ALM basada en P2P

(b) ALM no basada en P2P

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].

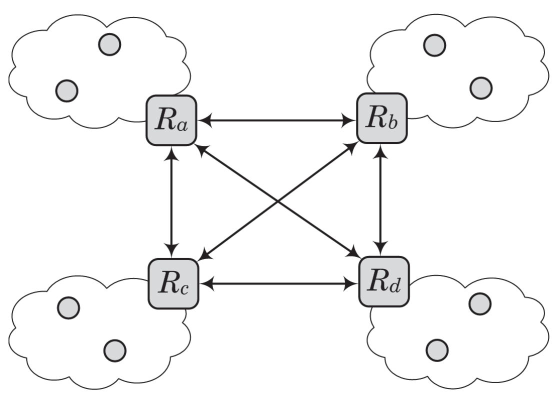


Figura 3.3: Overlay compuesta por reflectores.

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 su codificación o 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

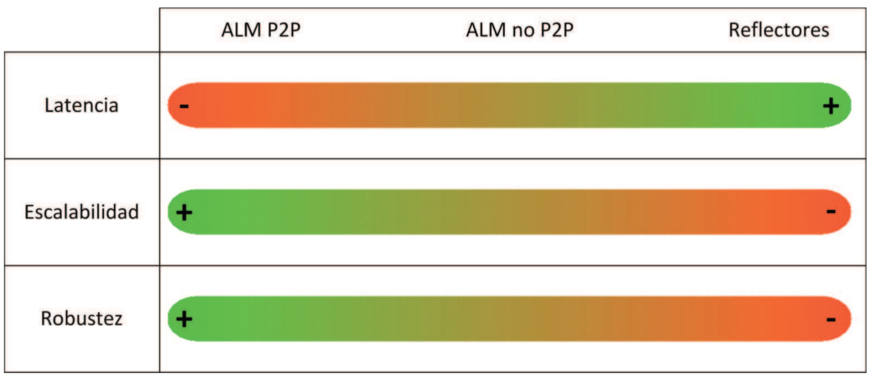


Figura 3.4: Comparativa entre las ALM y los reflectores.

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 autonómica (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 autonómicas 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 autonómicos

Como se ha visto anteriormente, las propiedades que componen inicialmente el paradigma de los sistemas autonómicos 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 autonómicas 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 autonómico es todo aquel que es capaz de auto-gestionarse, el paradigma de los sistemas autonómicos 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.

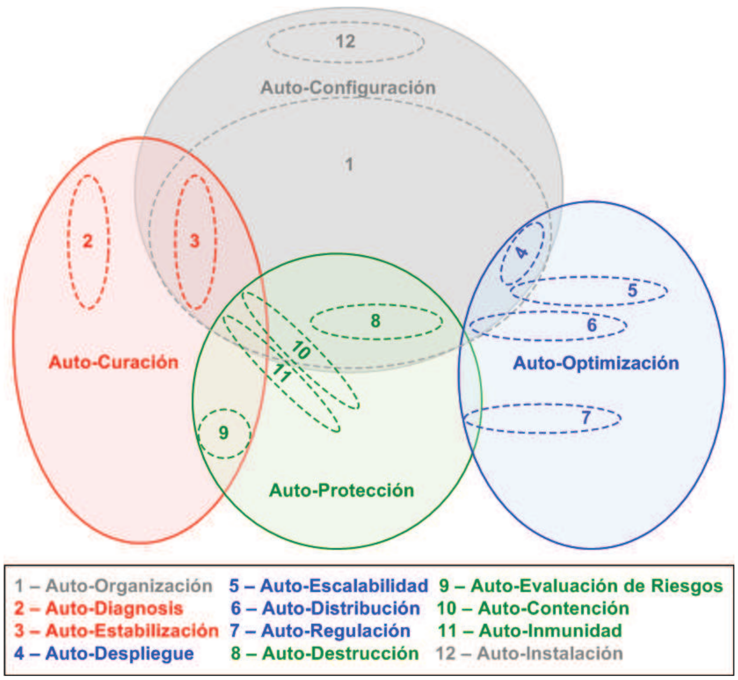


Figura 4.1: Subpropiedades autonómicas y su relación con las propiedades fundamentales.

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 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-diagnosis: 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-diagnosis, puesto que las acciones que lleva a cabo se realizan tras un diagnostico 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 satisface 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 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 ocasiones en las que para proteger al sistema puede ser necesario expulsar de la overlay a un miembro, 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 autonómico 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.

Tabla 4.1: Subpropiedades autonómicas en los SCM.

Trabajo Científico	**Overlay**	**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]	REF.	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	☐	☐
[Jannotti et al. 2000] [63]	ALM	☑	☑	☑	☑	☐	☐	☐	☐	☐	☑	☐	☐
[Pendarakis et al. 2001] [92]	ALM	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐	☐	☐
[Roca and El-Sayed 2001] [98]	ALM	☑	☑	☑	☐	☐	☐	☐	☐	☐	☑	☑	☐
[Banerjee et al. 2002] [8]	ALM	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐
[Chu et al. 2002] [22]	ALM	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐	☐
[Castro et al. 2003] [19]	ALM	☑	☑	☑	☑	☑	☑	☐	☐	☐	☑	☐	☐
[Hefeeda et al. 2003] [51]	P2P	☑	☑	☑	☑	☑	☑	☑	☐	☑	☐	☐	☐
[Luo et al. 2004] [73]	ALM	☑	☑	☑	☑	☐	☑	☑	☑	☑	☐	☐	☐
[Tran et al. 2004] [114]	ALM	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐
[Liu et al. 2005] [69]	ALM	☑	☐	☐	☑	☐	☐	☑	☐	☐	☐	☐	☐
[Zhang et al. 2005] [135]	ALM	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐
[Banerjee et al. 2006] [9]	ALM	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐
[Hales and Arteconi 2006] [46]	P2P	☑	☑	☑	☐	☐	☐	☐	☐	☐	☑	☐	☑
[Song et al. 2006] [108]	ALM	☑	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐
[Venkataraman et al. 2006] [118]	P2P	☑	☑	☑	☑	☑	☑	☐	☐	☐	☑	☐	☐
[Baccichet et al. 2007] [7]	P2P	☑	☑	☑	☑	☑	☑	☐	☐	☐	☑	☐	☐
[Maraviglia et al. 2007] [76]	REF.	☑	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	☐
[Pianese et al. 2007] [94]	P2P	☑	☑	☑	☑	☑	☑	☐	☑	☐	☑	☐	☐
[Zhu et al. 2007] [137]	P2P	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐	☐
[Mol et al. 2008] [80]	P2P	☑	☑	☑	☐	☐	☑	☐	☐	☐	☑	☐	☐
[Yin et al. 2008] [130]	P2P	☑	☑	☑	☑	☑	☑	☑	☐	☐	☑	☑	☐
[Liu et al. 2009] [70]	P2P	☑	☑	☑	☑	☑	☑	☑	☐	☐	☑	☐	☐
[Bikfalvi et al. 2010] [14]	P2P	☑	☑	☑	☑	☐	☐	☑	☐	☐	☑	☐	☐
[Ding et al. 2010] [27]	P2P	☑	☑	☑	☑	☑	☑	☑	☐	☐	☑	☐	☐
[Huang et al. 2010] [58]	ALM/P2P	☑	☑	☑	☑	☑	☑	☐	☐	☐	☑	☐	☐
[Payberah et al. 2010] [91]	P2P	☑	☑	☑	☑	☑	☑	☐	☐	☑	☑	☐	☐
[Qiu et al. 2010] [96]	P2P	☑	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐	☐
[Wang et al. 2010] [123]	P2P	☑	☑	☑	☑	☑	☑	☐	☐	☐	☑	☑	☐
[Wang and Chen 2011] [124]	ALM/P2P	☑	☑	☑	☑	☐	☑	☑	☐	☐	☑	☐	☐
[Granda et al. 2013] [42]	REF.	☑	☑	☑	☑	☐	☐	☑	☐	☐	☐	☐	☐

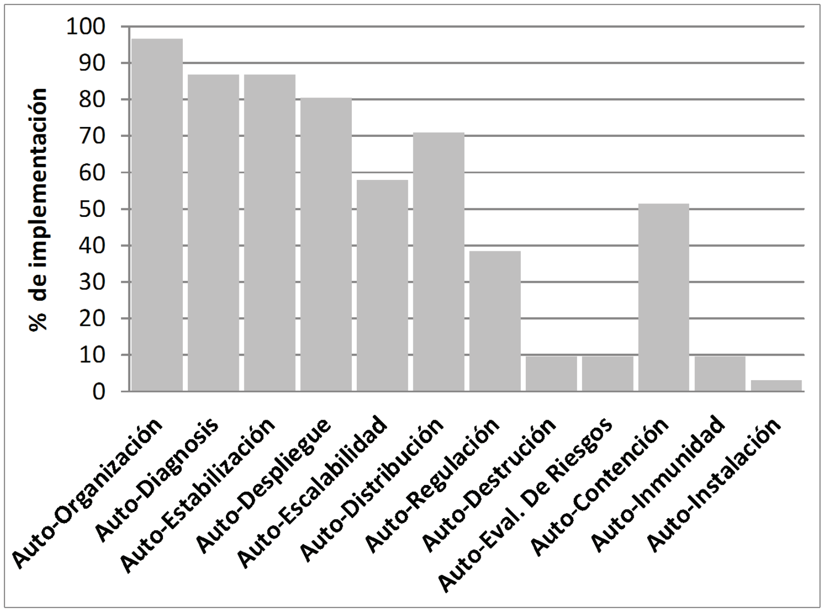


Figura 4.2: Porcentaje de implementación de cada subpropiedad autonómica.

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.

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).

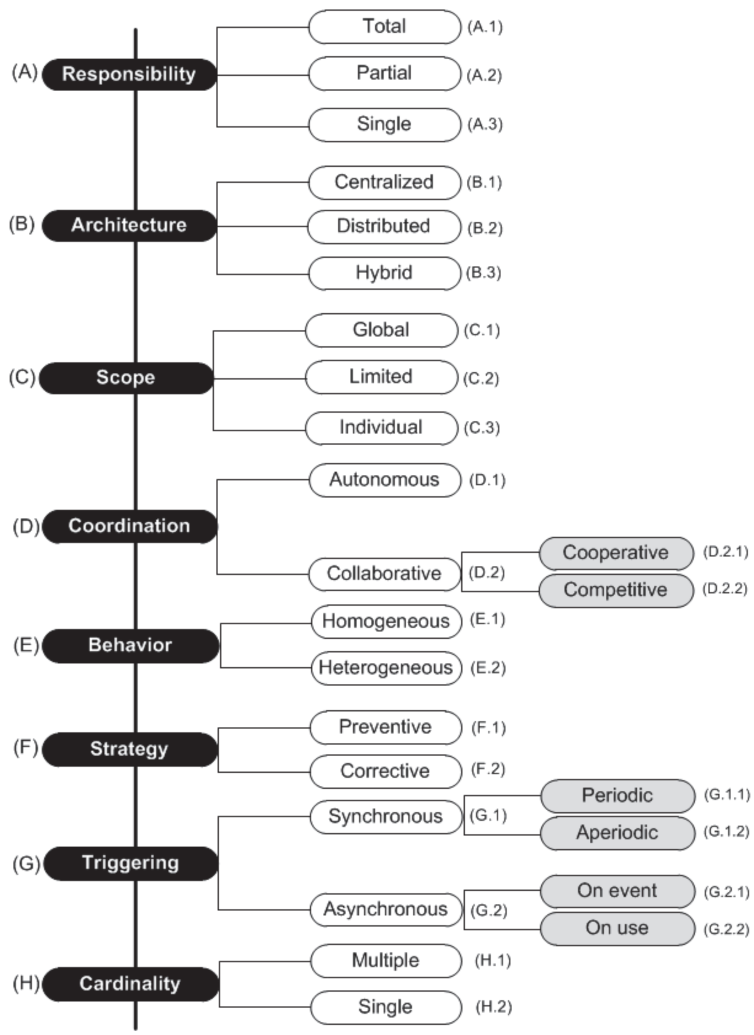


Figura 4.3: Taxonomía para clasificar las técnicas self-* en los SCM.

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).

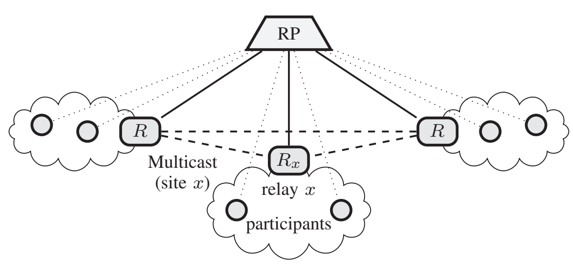


Figura 4.4: Overlay de la plataforma de distribución de la herramienta e-pSyLon.

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 RTCP[1] 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.

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.

[1] El 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.

Tabla 4.2: Estado de implementación de cada subpropiedad.

Subpropiedad	Implementada	Pendiente	No Aplicable
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	□	■	□

4.2.2. Propiedades autonómicas de la plataforma

El conjunto de subpropiedades autonómicas 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 lantecia 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.

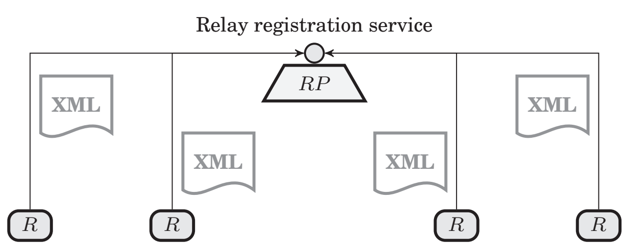


Figura 4.5: Primera fase de la técnica de auto-despliegue: registro de los reflectores.

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-diagnosis, 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.

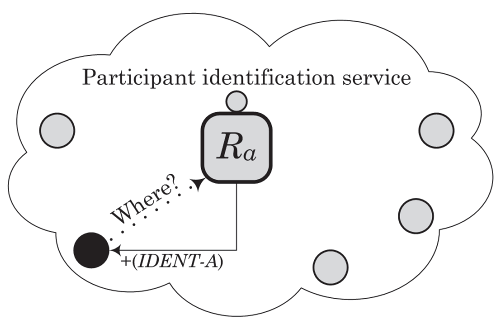


Figura 4.6: Segunda fase de la técnica de auto-despliegue: identificación de la sede corporativa.

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.

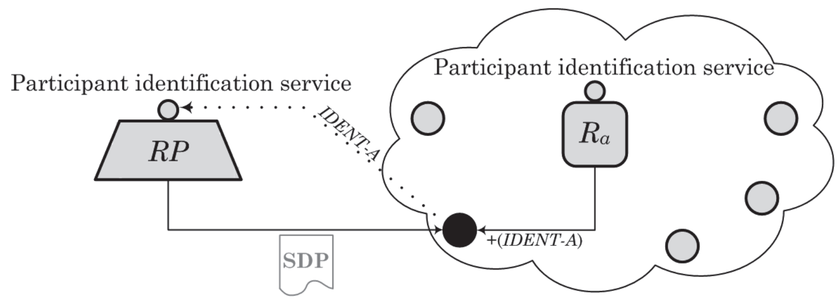


Figura 4.7: Tercera fase de la técnica de auto-despliegue: registro de los participantes.

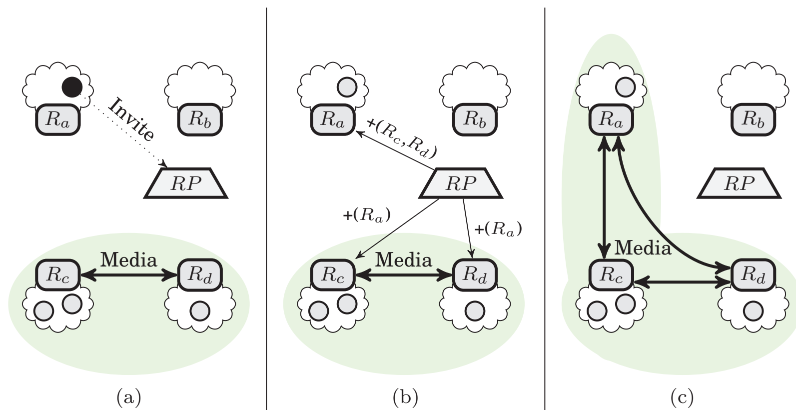


Figura 4.8: Inclusión de un reflector en la red virtual de reflectores.

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.

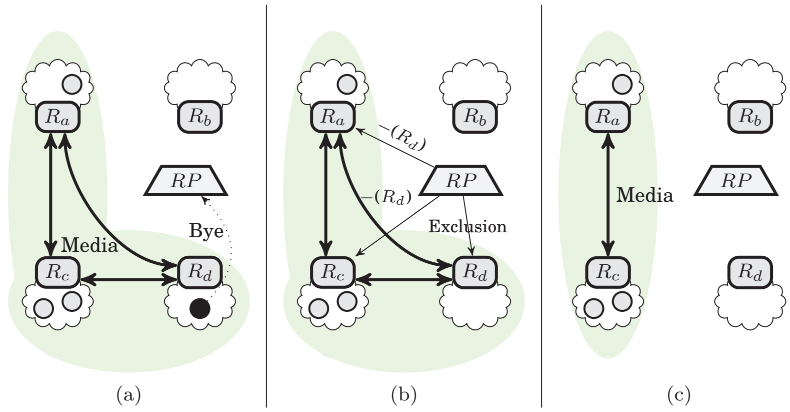


Figura 4.9: Exclusión de un reflector de la red virtual de reflectores.

Otras de las subpropiedades autonómicas implementadas en la plataforma son la auto-diagnosis 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.

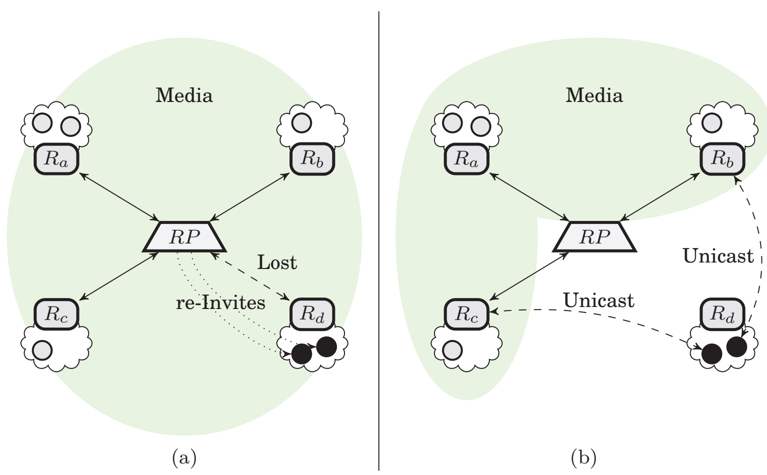


Figura 4.10: Técnica de auto-curación.

De manera más detallada, el proceso de auto-diagnosis 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.

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 conectivad 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) ← ∅
    for all participante ∈ Participantes(actividad) do
        reflector ← ReflectorOriginal(actividad, participante)
        if ActivarReflector(actividad, reflector) then
            sdp ← CrearSDP(actividad, reflector)
            Reinvitar(participante, sdp)
            Asociar(actividad, reflector, participante)
            ReflectoresActivos(actividad) ← reflector
        else
            Redirigir(actividad, reflector, participante)
        end if
    end for
end procedure

procedure Redirigir(actividad, reflector, participante)
    repeat
        nuevoReflector ← RedirigirA(Reflectores(actividad), reflector, participante)
    until ActivarReflector(actividad, nuevoReflector)
    sdp ← CrearSDP(actividad, nuevoReflector)
    Reinvitar(participante, sdp)
    Asociar(actividad, nuevoReflector, participante)
    ReflectoresActivos(actividad) ← nuevoReflector
    MarcarComoRedirigido(actividad, nuevoReflector, participante)
end procedure
```

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

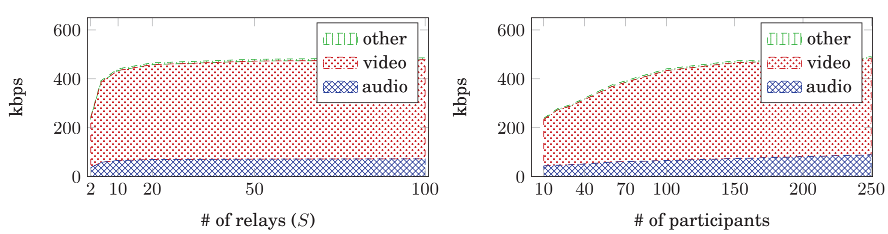
Tabla 4.3: Parámetros de la simulación a nivel de participante.

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)

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. Por su parte, 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 que se muestra 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 encuentran detallados en las publicaciones número 3 [43], y 8 [86] del compendio de publicaciones (secciones 6.1.3 y 6.2.1).

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.

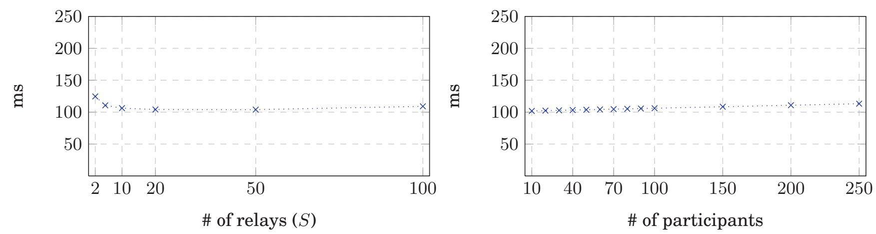


(a) 100 participantes (b) 10 reflectores

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.

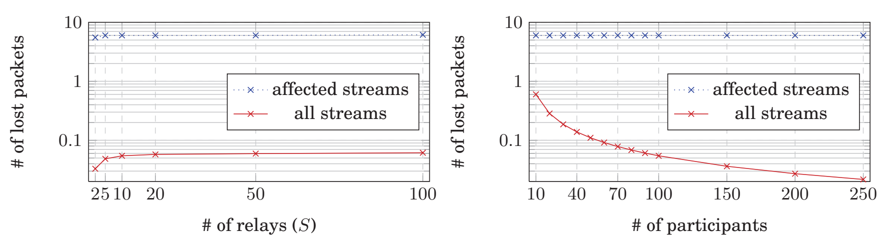
En la figura 4.13 se representa el tiempo empleado por la técnica de auto-curación para recuperar todos los participantes de una sede en la que se simula el error de su reflector correspondiente. El tiempo total de recuperación engloba los tiempos parciales empleados por las técnicas de detección del error (auto-diagnosis), 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.



(a) 100 participantes (b) 10 reflectores

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.



(a) 100 participantes (b) 10 reflectores

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

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.

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 intervencones y de flujos circulando por la plataforma, por lo que un error tiene menos incidencia en el promedio total.

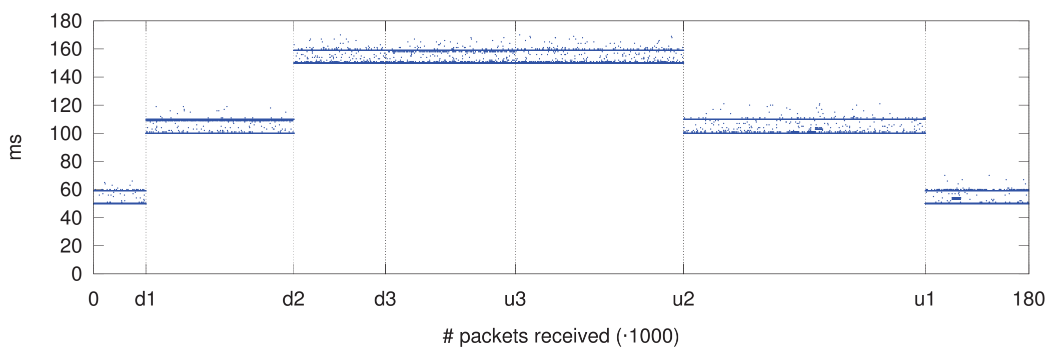


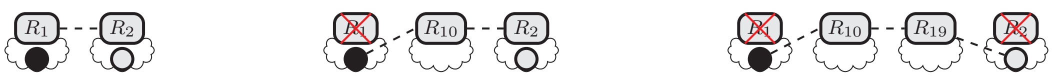
Figura 4.15: Evolución dinámica de la latencia

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



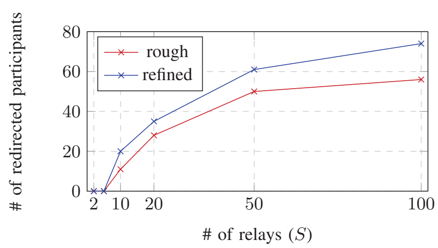
(a) (b) (c)

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

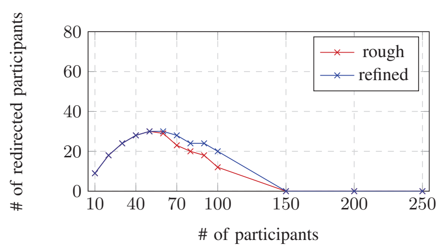
Cuando el reflector R_1 sufre un error (evento `d1`), el instructor es reubicado hacia el reflector R_{10}, por lo que la información multimedia debe atravesar dos enlaces unicast hasta alcanzar su destino tal y como muestra la figura 4.16b, lo que origina un incremento en la latencia. La latencia aumenta 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 considerse 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.



(a) Actividad con 100 participantes



(b) Actividad repartida en 10 sedes

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.

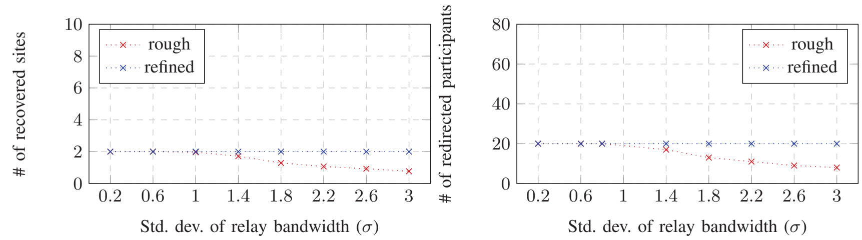
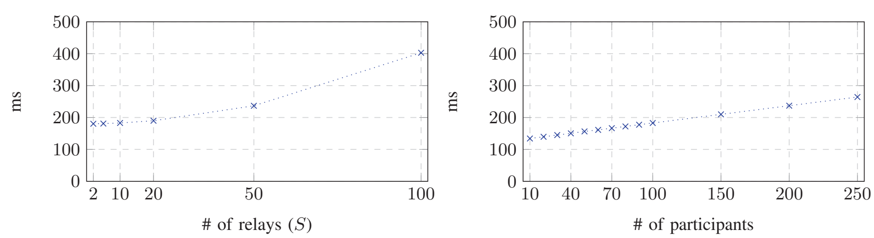


Figura 4.18: Número máximo de participantes e islas multicast estabilizados cuando los recursos de red no están balanceados.

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 compongan 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.



(a) Actividad con 100 participantes

(b) Actividad repartida en 10 sedes

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 autonómica 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 autonómica y la herramienta e-pSyLon

A continuación se presentan los resultados de utilizar en un entorno corporativo la plataforma de distribución autonómica 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.

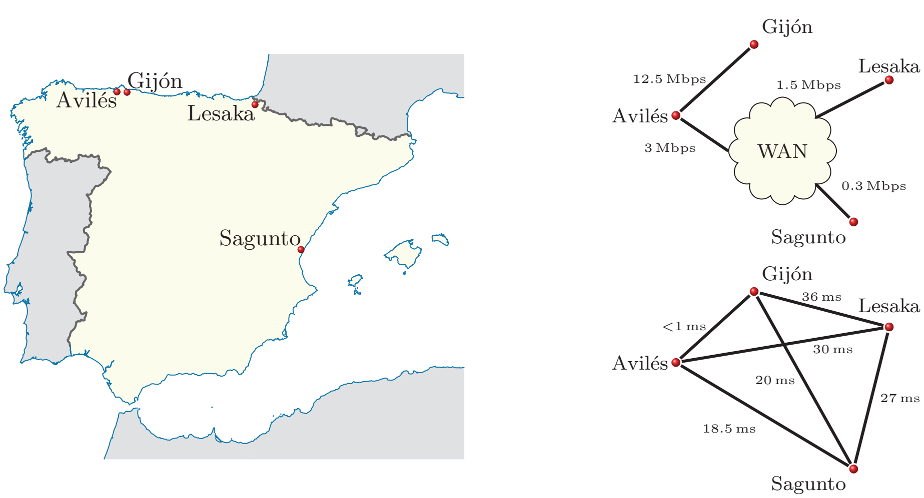


Figura 4.20: Localizaciones geográficas y ancho de banda disponible en el caso de uso de real.

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 bastante 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 las respuestas de los participantes, se puede extraer la conclusión de 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, 132]. 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 especificas [120]. Además, existen herramientas que combinan tanto el modelo online como la integración en herramientas de conferencia [78, 131].

La principal ventaja de este tipo de herramientas web es que reducen notablemente los problemas de compatibilidad, puesto que son desarrollos multiplataforma 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 *álter 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 Autonómica (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 autonómicas 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 autonómicas, 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 autonómica 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 autonómicas para desarrollar tareas de auto-gestión. Concretamente, subpropiedades autonómicas como la auto-organización, el auto-despliegue, la auto-diagnosis y la auto-estabilización son comunes en ese tipo de plataformas. Otras subpropiedades autonómicas, 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 autonómicas que faciliten su auto-gestión. Del estudio realizado se extrae también que las subpropiedades autonómicas 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 la 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-diagnosis, 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 dependencencia 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.

Computer Communications 36 (2013) 817–833

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Self-* in Multimedia Communication Overlays

Pelayo Nuño *, Juan C. Granda, Francisco J. Suárez, Daniel F. García

Department of Computer Science, University of Oviedo, Campus de Viesques s/n, 33204 Gijón, Spain

ARTICLE INFO

Article history:
Received 23 November 2011
Received in revised form 31 July 2012
Accepted 30 December 2012
Available online 7 January 2013

Keywords:
Autonomic computing
Multimedia Communication Systems
Overlays
Self-*
Taxonomy

ABSTRACT

Multimedia Communication Systems (MCSs) usually deploy complex data distribution overlays, making the efficient interchange of multimedia content possible. The time constraints imposed by interactive communications further complicate the management of the data distribution overlays. Many MCSs have the ability to automatically manage the overlay, hence they behave as autonomic computing systems, facing events that imply changes in the relations or in the performance of their members. Self-management characteristics related to organizing the members of the data distribution overlays, balancing resources, and aggregating efforts are present in many MCSs. These characteristics, or self-properties, can be achieved using various techniques, according to the complexity and time constraints of each MCS. The aim of this paper is to classify the techniques implemented in MCSs to achieve self-management characteristics in their overlays. These techniques are classified according to a taxonomy proposed in this work, providing a unified and practical framework for understanding and comparing them.

1. Introduction

The term autonomic computing was coined in 2001 by Paul Horn, IBM senior vice president and director of research, at the Academy of Engineering of Harvard University. The purpose of his manifesto was to alert the scientific community and the IT industry to the increasing complexity of computing systems, and the looming crisis this entailed [1].

The aim of autonomic computing systems is self-management, *i.e.* the ability of a system to manage itself while hiding its complexity to users [1–4]. Autonomic systems adjust their behavior according to changes in their components or in their environment, and to prevent external threats. The goal of self-management systems is twofold; they hide the details of operating and maintaining the system from administrators, and provide customers with a reliable system that is operative at its highest throughput at any time. Thus, the users can focus on what they expect from the system, not on how the service is provided [5].

Nowadays, MCSs are widely present in areas such as industry, business, education or leisure. MCSs can be classified in content distribution, on-demand media streaming, live media streaming and conferencing MCSs. All these types of MCSs deploy data distribution overlays among users to cope with latency, heterogeneity, scalability, availability and confidentiality issues. They are composed of many related entities providing services such as authentication, data distribution, session identification, etc. The self-management characteristics present in modern MCSs to manage their overlays make them good examples of autonomic computing systems.

Content distribution MCSs transfer multimedia files from a source to one or various receivers. Each receiver may become a source of multimedia files saving the original source from retransmissions. On-demand and live media streaming MCSs deliver data in media streams. Media is produced and delivered on the fly in the latter, while pre-recorded media is used in the former. This imposes stricter temporal constraints in live media streaming MCSs. Unlike live MCSs, on-demand media streaming MCSs usually allow receivers to move the playback position forward and backward the multimedia streams. Thus, on-demand MCSs react to these operations to discover the source that must serve the multimedia data at the playback position. Finally, conferencing MCSs must be prepared for real-time interactive communications. Conceptually, these MCSs are similar to live media streaming MCSs but impose even more severe temporal constraints.

This paper identifies the autonomic characteristics of MCS overlay management, and the self-* techniques that are commonly used. Furthermore, a taxonomy is proposed in order to classify these techniques. To the best of the authors' knowledge, this research is the first to cover these aspects of autonomic MCSs overlays.

The remainder of the paper is organized as follows. Section 2 introduces MCS overlays. The basic self-properties of autonomic

* Corresponding author. Address: Office 1.S.03, Edif. Departamental Oeste, Campus de Viesques, 33204 Gijón, Spain. Tel.: +34 985 18 22 27; fax: +34 985 18 19 86.

E-mail addresses: nunopelayo@uniovi.es (P. Nuño), jcgranda@uniovi.es (J.C. Granda), fjsuarez@uniovi.es (F.J. Suárez), dfgarcia@uniovi.es (D.F. García).

0140-3664/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.12.009

computing included in MCSs are briefly introduced and further divided into subproperties in Section 3. A taxonomy to classify techniques to achieve such subproperties is proposed in Section 4. An analysis and classification of the techniques identified in MCSs according to the presented taxonomy is carried out in Section 5. Finally, Section 6 contains the concluding remarks.

2. Multimedia Communication Overlays

Overlays are the underlying mechanisms enabling multipoint communication in MCSs [6]. Overlays are complex transmission systems deployed on top of actual networks to transport real-time multimedia content to many concurrent users under several constraints. They must operate over a variety of networks, facing heterogeneous issues, minimizing end-to-end latency and jitter, and using available resources efficiently while maintaining a reasonable level of quality of service. Thus, the design of the MCS overlays should be guided by two commonly conflicting objectives: low latency and high scalability. Different latency constraints on communications are imposed depending on the type of MCS, which are relaxed for content distribution, strict for on demand and live streaming MCS, and highly strict for conferencing MCSs. Conversely, the scalability requirements for content distribution and media streaming MCSs are usually higher than those for conferencing MCSs.

Several surveys about overlays implementing group communication services have been developed [6,7]. Overlays are divided into different classes, although three classes are usually considered for developing time-constraint communication solutions: unicast/multicast proxies (*a.k.a.* reflectors), automatic overlay multicast (*a.k.a.* Application Layer Multicast, ALM) and peer-to-peer (P2P). A reflector is a network proxy that forwards incoming multimedia traffic to one or many entities, which may be endpoints or other reflectors. Thus, it is possible to deploy an overlay based on reflectors, avoiding inflicting routing and management tasks on the rest of the overlay members. On the other hand, data delivery between members and other management tasks are usually supported by every member in ALM and P2P overlays. The main difference between them is the knowledge about the composition of the overlay; fully known by a central entity, by all the members, or distributed in ALM overlays, and incremental and propagated in P2P overlays.

Some authors further classify overlays in structured and unstructured overlays. Structured overlays are built according to a fixed topology and data is delivered through such topology. In contrast, data is divided into chunks that are spread across the overlay over highly volatile topologies in unstructured overlays.

Structured overlays can be deployed according to several topologies as illustrated in Fig. 1. The most common are trees and meshes. Several kinds of tree topologies are possible (Fig. 1(a)), such as shortest path trees and minimum spanning trees, focusing on minimizing latency and node degree respectively. Meshes are graph topologies that can be interconnected in different ways. All members are connected to a central member in a star topology (Fig. 1(b)). In contrast, every member in the mesh is connected to each other in a full-mesh topology (Fig. 1(c)). Each member of a ring topology is connected exclusively to another two members (Fig. 1(d)). A sun topology is like a star topology, but with a non-null diameter, that is, the backbone of the topology is a mesh-based subset of the overlay rather than a single member. For example, a sun topology with a ring backbone is illustrated in Fig. 1(e). Finally, hybrid topologies (Fig. 1(f)) are possible combining tree and mesh topologies in multiple ways. For example, members are organized in various local meshes across a global distribution tree in a clustered topology.

The topology of MCS overlays has enormous implications for the scalability and the latency of communications. Both tree and star topologies are appropriate for MCSs with a single source. Those MCSs with strict temporal constraints must use topologies with a high node degree (star or full-mesh) since the number of hops is minimized. Topologies such as trees or hybrid topologies must be adopted to make scalability possible at the expense of higher latency when the overlay is composed of a large number of members. Full-mesh topologies are appropriate for overlays where each member can act as a data source. Sun and ring topologies are suited to MCSs with high scalability requirements and without strict real-time constraints. Finally, hybrid topologies have more flexibility and scalability, but at the expense of increasing complexity and latency. They are appropriate for heterogeneous MCSs where unicast and multicast delivery are combined.

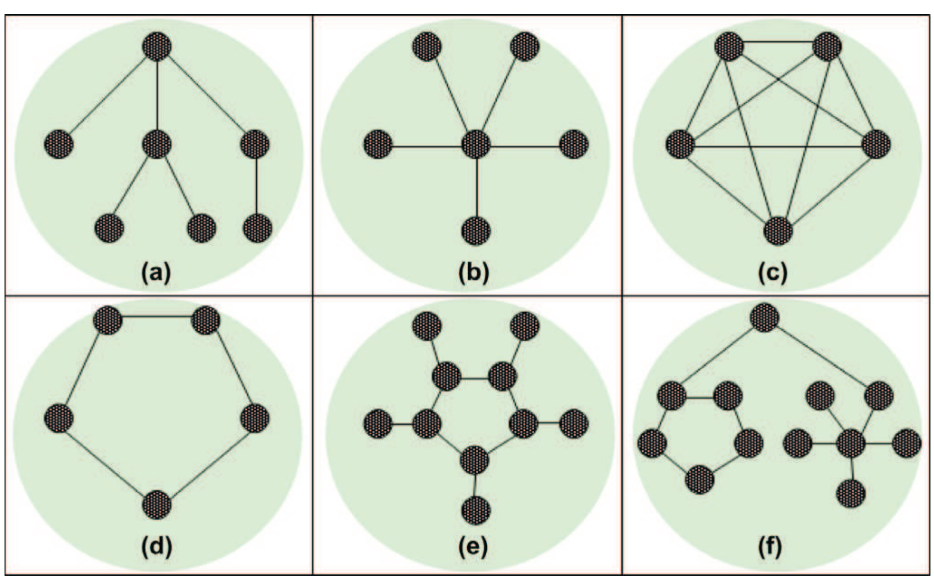


Fig. 1. Overlay topologies: (a) tree, (b) star, (c) full-mesh, (d) ring, (e) sun, (f) hybrid.

MCS overlays based on reflectors usually build a sun topology where each member of the backbone is a reflector connected with the others according to a full-mesh topology. Therefore, minimum latency is achieved, since the number of hops is limited to two reflectors maximum. However, the scalability of this approach is limited as data traffic increases exponentially as the number of reflectors grows. The overlay is managed by a central entity that may also act as entry point, which is known in the literature as Rendezvouz Point (RP). In contrast, MCSs based on ALM or P2P admit a wider topology variety. ALM and P2P are flexible overlays where connections between members can be dynamically modified to obtain optimal routing paths. These overlays can be deployed in mesh, tree and multi-tree and hybrid topologies dividing the overlay in clusters, or building tree-mesh and multi-tree-mesh topologies. In this case, ALM overlays can be managed in a centralized or in a distributed manner, while P2P-based are managed distributedly. Nevertheless, in spite of achieving an efficient data delivery, the latency introduced in communications may be high, as data traverses many hops from the source to receivers.

All types of group communication services can be found in content distribution MCS overlays. Reflector overlays can be deployed following structured tree [8] or mesh [9] topologies. ALM overlays can be deployed following structured full-mesh [10], tree [11] and multi-tree topologies [12], or cluster based topologies [13,14]. P2P overlays can be deployed according to an unstructured mesh [15].

Only ALM and P2P overlays are used in on-demand media streaming MCSs due to scalability requirements. Structured topologies such as trees [16] and multi-trees [17] are possible for ALM overlays. P2P overlays can be deployed in structured mesh topologies [18,19], or unstructured highly volatile mesh [20] and multi-tree [21] topologies.

Live media streaming MCSs are also exclusively based on ALM and P2P overlays. Structured overlays can be deployed according to tree [22] topologies in ALM overlays, multi-tree topologies both in ALM [23] and P2P [24] overlays, and also according to clustered hybrid topologies both in ALM [25,26] and P2P overlays [27]. On the other hand, unstructured P2P overlays can be deployed according to mesh [28–30] and highly volatile multi-tree [31,32] topologies. There are also overlays combining both structured and unstructured features using tree-mesh topologies [33].

Finally, ALM and reflector overlays are mainly used in conferencing MCSs due to the strict latency constraints imposed. ALM overlays are suitable for conferences with a high number of participants, while reflector overlays are ideal for low latency conferencing. Many-to-many communications are possible using either ALM overlays and multi-tree topologies [34,35], with a tree rooted in each source, or reflector overlays deployed in tree [36], or full-mesh topologies [37] provided that the number of participants is low. One-to-many communication can be deployed through clustered overlays [38].

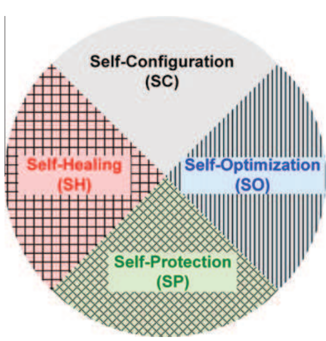


Fig. 2. Fundamental self-properties of autonomic computing systems.

3. Autonomic computing self-properties

The main purpose of an autonomic computing system is self-management. Autonomic systems exhibit the characteristics depicted in Fig. 2, usually referred to as self-properties [2–4].

- Self-configuration (SC) refers to the ability of a system to adapt dynamically and automatically to the environment, and to changes in it.
- Self-healing (SH) systems are capable of identifying, diagnosing and recovering from unforeseen interruptions and errors, in both software and hardware, that compromise their performance.
- Self-optimization (SO) refers to the ability of a system to adjust the use of resources for maximum efficiency and performance.
- A self-protected (SP) system provides proactive and/or reactive mechanisms to be secure, resilient and fault-tolerant, minimizing the effects of attacks and cascade failures.

These properties can be understood as fundamental objectives of the autonomic systems. New self-properties have been proposed recently as research into autonomic computing progresses, increasing the initial set of four self-properties. However, we consider that this wide range of new self-properties differs from the initial self-properties. The new self-properties can be considered as autonomic behaviors to accomplish the desired objectives, that is, these new self-properties act as pillars to reach one or several initial self-properties. In this work such pillars are referred to as *self-subproperties*. Therefore, self-properties can be decomposed into one or several subproperties, which in some cases permit relations between self-properties to be established [39]. Self-subproperties can be defined as follows. Let $S*$ be the set of self-properties and let Q be the set of self-subproperties, then:

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

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

Lemma 1. *An autonomic computing self-subproperty, single or combined with others, consists in a behavior expected in a self-property.*

Thus, a self-subproperty can be understood as an autonomic behavior or capability from which a self-property emerges.

3.1. Self-Subproperties in Multimedia Communication Systems

A classification of autonomic properties has not been globally established. Although other classifications are possible, we classify self-subproperties according to their implications to MCS overlay management. Fig. 3 shows the self-subproperties of MCSs.

Self-organization is the ability of a system to interact with the environment and other systems to achieve objectives exceeding its capabilities as a single entity, without being guided or managed by an external entity [40]. Self-organization is clearly a characteristic of self-configuration systems. The objective of changing the organization of a system may be to improve, restore or maintain one or several features [5]. Self-organization is a quite common feature in MCSs to improve their overall performance, or face failures, by modifying network connections among the overlay members. Therefore, self-organization leads to optimizing, healing or protecting MCS overlays.

Self-diagnosis refers to the ability of a system to identify malfunctioning parts and localize and address the cause of these failures [41]. A correct diagnosis is mandatory to perform self-healing actions. In other words, self-healing relies on self-diagnosis, so

Fig. 3. Self-subproperties related to self-properties.

this self-subproperty can be considered a constituent characteristic of self-healing systems. Therefore, every self-healing system must be self-diagnosable [42]. This self-subproperty is widely implemented in MCS overlays, so members are able to detect network disruptions with their peers in the overlay. The way of accomplishing such detections highly depends on the communications protocol.

Self-stabilization, also known as self-recovery, is the automatic response of a system facing failures to achieve a steady state from any arbitrary configuration [43]. Other authors consider a system as self-stabilized if it moves to an active and steady state and keeps its availability after the occurrence of a failure affecting it or its environment [44]. Self-stabilization is the complementary action to self-diagnosis for recovering a system completely. This self-subproperty always relies on self-organization and hence on self-configuration in MCS overlays. An overlay recovers from disruptions by establishing new relations among the affected members and other members within the overlay.

A *self-deployable* system is able to incrementally build the relations among members based on initial directives, generating a full operative environment that satisfies a deployment goal [45]. *Self-deployment* can be considered a special case of self-stabilization, representing an evolution from an initial state. This similarity is especially appreciable when the system is deployed without using information from previous deployments, which is known as an oblivious model [46]. Self-deployment is associated with self-organization and self-optimization rather than with self-healing, since it is focused on optimizing a deployment objective. Self-deployment copes with managing the joining requests from incoming members and their initial placement in the overlay.

A *self-scalable* system performs properly and adapts automatically to the demand requested by other systems or users [47]. Self-scalable MCSs are not compromised facing the growth of the overlays. Self-scalability is mainly related to self-optimization [48], although some relations with self-organization may also exist [49] since a scaling strategy may imply variations in the organization of the overlay.

Self-distribution refers to the ability of a system to automatically determine the dependability of components, schedule and balance the workload among them, and allocate and coordinate resources according to system demands [50]. Resource allocation, tuning resources and automatic workload management are well-known features of self-optimizing systems [3], so the relation between self-distribution and self-optimization is clear. Furthermore, the application of self-distribution techniques may lead to changes in the organization of the system, so self-distribution is also associated with self-organization. MCS overlays balance the workload by distributing the data delivery effort among data paths equally. This self-subproperty is present in those MCSs where source members are able to operate in a combined manner. Similarly, those MCSs with distribution strategies promoting a balanced data dissemination also meet self-distribution.

Self-regulation is the ability of a system to adjust or reconfigure its parameters to guarantee the quality of service depending on the operational conditions and the environment [51]. Self-regulation is related to self-optimization, satisfying system-wide properties such as fairness among concurrent applications and clients, or critical application requirements such as preserving a level of performance [52]. Maintaining a level of performance also protects the system preventing any component from reaching an excessive level of utilization. In addition, self-regulation techniques may improve the security of MCSs, avoiding malicious uses [53]. Thus, self-regulation is also related to self-protection. In the context of MCSs, self-regulation can be identified in those systems with members that are able to modify both the media encoding and the composition of the content delivered across the overlay on the fly. Thus, the presence of bottlenecks and overloads in the overlay is minimized. Rate streaming control and scalable video coding are examples of self-regulation mechanisms usually employed in MCSs.

Self-destruction refers to the ability of a system to abort execution when an unsolvable safety problem or behavior that might damage the system or the environment (including users) is detected [54]. This leads to an increase in the overall protection of the system. Self-destruction can be considered as the opposite of self-deployment in MCSs, *i.e.* stopping a media delivery or excluding a member, so it may also imply a reorganization of the system. This self-subproperty can be observed in MCSs where specific resource-consuming media data can be discarded, for example stopping video streams and forwarding exclusively audio streams. Self-destruction promotes discarding what is not essential in order to ensure efficiency.

Self-assessment of risks represents the ability of a system to determine and evaluate privacy or security vulnerabilities, exploits and service disruptions. The system is able to predict not only a particular risk, but effects, cascade reactions, and future implications of the actions performed to mitigate such risk [55]. Hence, an autonomic system can behave in a proactive or reactive manner, which leads to self-protection or self-healing implications respectively [56]. In a proactive manner, self-assessment of risks can be considered as an anticipated self-diagnosis, as current risks that might lead to future failures can be detected. This implies that the system is not healed but it is protected. In the context of MCSs, this self-subproperty is implemented in those systems where members are able to estimate the impact of an operational circumstance both in the performance of the overlay and the quality of service observed by users.

Self-containment systems are able to minimize the impact of malicious actions and failures, affecting only a part of the system, while its operativeness is restored with the non-compromised components [39]. Self-containment is mainly presented as a form of self-protection. However, the capability of restoring its operativeness also implies a link with self-healing. Furthermore, a self-organization process may be required to achieve the containment objective. MCSs exhibit self-containment when they are able to isolate those members that compromise the overall performance of the overlay. For example, if the connectivity of a member be-

Table 1
Self-subproperties of MCSs.

Research Work	Overlay type	Self-Organization	Self-Diagnosis	Self-Stabilization	Self-Deployment	Self-Scalability	Self-Distribution	Self-Regulation	Self-Destruction	Self-Asses. Risks	Self-Containment	Self-Immunity	Self-Installation
[Chawathe et al. 2000][9]	REF.	☐	☐	☐	☐	☐	✔	✔	☐	☐	☐	☐	☐
[Jannotti et al. 2000][16]	ALM	✔	✔	✔	✔	☐	☐	☐	☐	☐	✔	☐	☐
[Pendarakis et al. 2001][11]	ALM	✔	✔	✔	✔	☐	☐	☐	☐	☐	☐	☐	☐
[Roca and El-Sayed 2001][8]	ALM	✔	✔	✔	☐	☐	☐	☐	☐	☐	✔	✔	☐
[Banerjee et al. 2002][13]	ALM	✔	✔	✔	✔	✔	✔	☐	☐	☐	☐	☐	☐
[Chu et al. 2002][10]	ALM	✔	✔	✔	✔	✔	☐	☐	☐	☐	☐	☐	☐
[Castro et al. 2003][12]	ALM	✔	✔	✔	✔	✔	✔	☐	☐	☐	✔	☐	☐
[Hefeeda et al. 2003][18]	P2P	✔	✔	✔	✔	✔	✔	✔	☐	✔	☐	☐	☐
[Luo et al. 2004][35]	ALM	✔	✔	✔	✔	☐	✔	✔	✔	✔	☐	☐	☐
[Tran et al. 2004][25]	ALM	✔	✔	✔	✔	✔	✔	☐	☐	☐	☐	☐	☐
[Liu et al. 2005][34]	ALM	✔	☐	☐	✔	☐	☐	✔	☐	☐	☐	☐	☐
[Zhang et al. 2005][28]	ALM	✔	✔	✔	✔	✔	✔	☐	☐	☐	☐	☐	☐
[Banerjee et al. 2006][22]	ALM	✔	✔	✔	✔	✔	✔	☐	☐	☐	☐	☐	☐
[Hales and Arteconi 2006][15]	P2P	✔	✔	✔	☐	☐	☐	☐	☐	☐	✔	☐	✔
[Song et al. 2006][38]	ALM	✔	☐	☐	☐	☐	✔	☐	☐	☐	☐	☐	☐
[Venkataraman et al. 2006][31]	P2P	✔	✔	✔	✔	✔	✔	☐	☐	☐	✔	☐	☐
[Baccichet et al. 2007][24]	P2P	✔	✔	✔	✔	✔	✔	☐	☐	☐	✔	☐	☐
[Maraviglia et al. 2007][36]	REF.	✔	☐	☐	☐	☐	☐	✔	✔	☐	☐	☐	☐
[Pianese et al. 2007][29]	P2P	✔	✔	✔	✔	✔	✔	☐	✔	☐	✔	☐	☐
[Zhu et al. 2007][14]	P2P	✔	✔	✔	✔	✔	✔	☐	☐	☐	☐	☐	☐
[Mol et al. 2008][21]	P2P	✔	✔	✔	☐	☐	✔	☐	☐	☐	✔	☐	☐
[Yin et al. 2008][27]	P2P	✔	✔	✔	✔	✔	✔	✔	☐	☐	✔	✔	☐
[Liu et al. 2009][30]	P2P	✔	✔	✔	✔	✔	✔	✔	☐	☐	✔	☐	☐
[Bikfalvi et al. 2010][17]	P2P	✔	✔	✔	✔	☐	☐	✔	☐	☐	✔	☐	☐
[Ding et al. 2010][20]	P2P	✔	✔	✔	✔	✔	✔	✔	☐	☐	✔	☐	☐
[Huang et al. 2010][26]	ALM/P2P	✔	✔	✔	✔	✔	✔	☐	☐	☐	✔	☐	☐
[Payberah et al. 2010][32]	P2P	✔	✔	✔	✔	✔	✔	☐	☐	✔	✔	☐	☐
[Qiu et al. 2010][19]	P2P	✔	✔	✔	✔	✔	✔	✔	☐	☐	☐	☐	☐
[Wang et al. 2010][33]	P2P	✔	✔	✔	✔	✔	✔	☐	☐	☐	✔	✔	☐
[Granda et al. 2011][37]	REF.	✔	✔	✔	✔	☐	☐	✔	☐	☐	☐	☐	☐
[Wang and Chen 2011][23]	ALM/P2P	✔	✔	✔	✔	☐	✔	✔	☐	☐	✔	☐	☐

comes unstable, or penalizes the performance of the overlay, the member may be relocated in the overlay to minimize the negative impact.

A *self-immune* system restores safety and reliability after the first occurrence of an event, which no longer compromises the system during its activity period. Two lemmas have been proposed to explain the relation between self-immunity and self-healing, and between self-immunity and self-protection [39]:

1. Every self-immune system is self-healing, but the opposite is not true.
2. Every self-protected system is self-immune, but the opposite is not true.

MCSs implement self-immunity in conjunction with self-containment to cope with several issues, such as the presence of suspicious members. Some MCSs ensure the overlay performance by placing those members with high probability of causing an overlay partition at the edge of the overlay. This action limits the impact of a failure to exclusively the faulty member as no other members are descendant of the former, which means that the system is able to minimize the consequences of such failure (self-containment) along the ongoing session (self-immunity). Encryption, message authentication and integrity checks are other forms of self-immunity in MCSs to face unauthorized access from untrusted members.

Self-installation is the ability of a system to add new components and software, or reinstall programs after crashes or degradations [54]. It can be considered a characteristic of self-configuration systems. Other terms related to self-installation are self-calibration of installed elements and their updates [57]. Although this feature is still not widespread, it can be observed in some MCSs where members obtain media codecs during the deployment phase, or even modify their operation mode on the fly by copying behaviors or functional strategies from other overlay members.

Self-management is an inherent part in MCS overlays to cope with their performance targets, and that fact makes MCS overlays interesting examples of autonomic computing systems. The entities composing an overlay can be distributed and act independently, without awareness about the whole system so they must face their management by themselves, showing neither changes in their behavior nor degradation in the quality of service to their users. Table 1 depicts the autonomic characteristics of several prominent MCSs that have been analyzed.

4. A taxonomy of techniques achieving self-subproperties

Many MCSs exhibit autonomic properties when managing their overlays. They implement self-management techniques to face challenges such as reorganizing the overlay to increase performance, guarantee QoS, and cope with member failures. The differences among techniques arise from the characteristics of MCSs and their purpose. Thus, MCSs oriented to few-to-few real-time communications differ widely from other MCSs focused on one-to-many or many-to-many content distribution. They differ in aspects such as: centralized or distributed overlay maintenance,

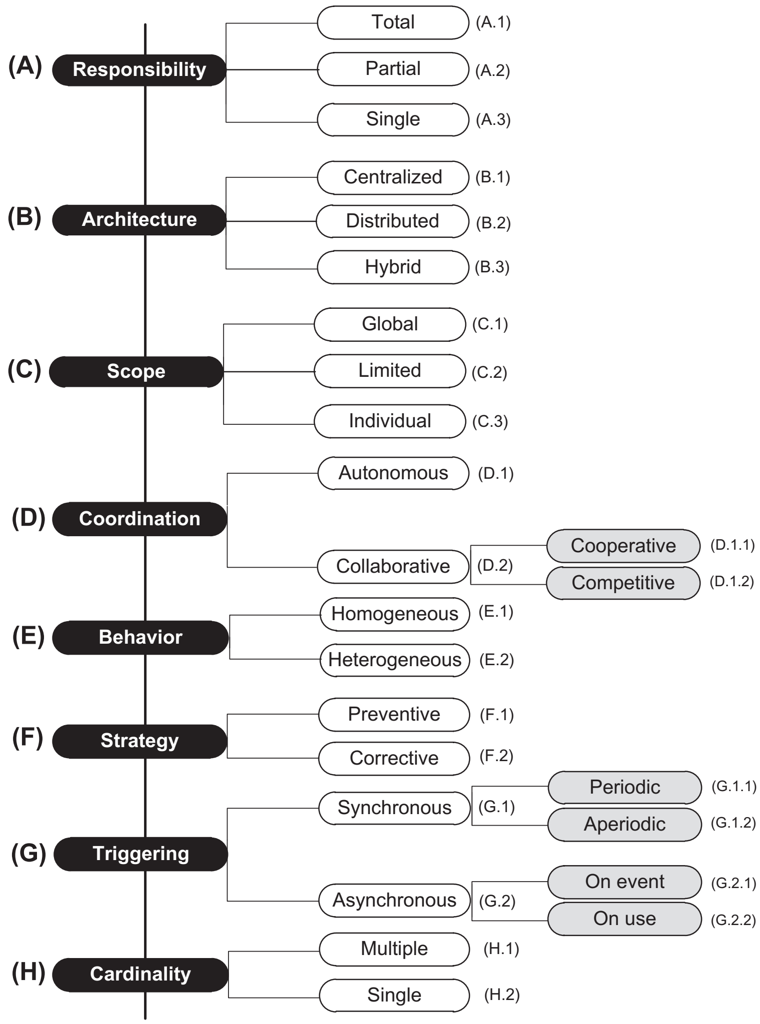


Fig. 4. Taxonomy of self-* techniques in MCS.

mesh-first or tree-first overlay construction, or the data delivery approach.

An autonomic self-subproperty can be achieved using various techniques. A taxonomy of these techniques is presented in Fig. 4, enabling the comparison of techniques achieving a specific self-subproperty. Other taxonomies have been proposed addressing issues related to autonomic computing, both in general terms [58], grid computing environments [59], and the nature of self-properties [60]. However, to the best of the authors' knowledge, this research is the first to analyze the autonomic techniques in MCSs. It should be noted that we do not claim this taxonomy to be complete, indeed, further additions are expected. Future research might increase the range of criteria through new approaches to cope with similar issues or new challenges.

Several criteria should be analyzed for describing techniques implemented in MCS overlays properly. Most of these criteria can be expressed algebraically, so some definitions are introduced. Let Q be the set of self-subproperties, M the members composing a reference overlay S, and T a set of techniques that are able to ensure a specific self-subproperty q.

The first characteristic of techniques to achieve self-subproperties in MCSs that can be identified is the responsibility, which refers to the part of the overlay responsible for performing the technique. The **responsibility** of the technique can be total, partial or single. Fig. 5 shows three overlays composed of several members, where responsible members are highlighted with a pattern fill. Total-responsibility techniques require that all the members (sometimes except the source) perform the technique as shown in Fig. 5(a) (A.1). Many self-organization techniques of P2P overlays are classified as total-responsibility, since all the members of the overlay are involved in the reorganization decisions. The techniques where responsibility relies on a subset of the overlay members are classified as partial-responsible (A.2), as Fig. 5(b) illustrates. Finally, the techniques where responsibility relies on a single member of the overlay are classified as single-responsibility as presented in Fig. 5(c) (A.3). To formally define responsibility, let $N_t \subseteq M$ be the responsible members for a technique t, then:

$$t \in \text{A.1} \iff N_t = M$$

$$t \in \text{A.2} \iff |N_t| > 1 \wedge N_t \neq M$$

$$t \in \text{A.3} \iff |N_t| = 1$$

The responsible members for a self-* technique must make decisions. They organize according to an **architecture** to make decisions. Centralized techniques are managed by a single overlay

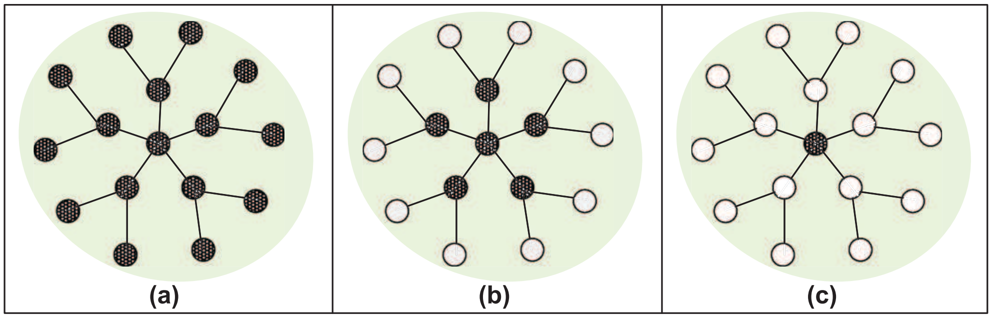


Fig. 5. Types of responsibility for a self-* technique in MCSs: (a) total, (b) partial, (c) single.

member (B.1), such as in self-organized overlays with an RP. A single-responsibility technique always has a centralized architecture, but the opposite is not true. For example, management decisions may be taken centralizedly according to reports from other members in self-organized overlays. Thus, the technique is performed by several members of the overlay but a single member makes the reorganization decisions. In contrast, many overlay members make decisions using distributed techniques (B.2). This is common in self-organized P2P MCSs implying continuous interactions among the members of the overlay. Finally, hybrid techniques combine both centralized and distributed approaches, building a multilevel decision architecture (B.3). Some overlays with sun topologies use techniques that organize the backbone distributedly, but each member of the backbone also manages a segment of the overlay according to a centralized architecture.

The responsible members for a self-* technique have to interact with other members of the overlay. The **scope** of a technique refers to the members with which a responsible member has to interact. Techniques where the responsible members have to interact with the whole overlay have global scope (C.1). This usually occurs in overlays built according to a star topology, where the central entity has to interact with all the overlay members to carry out reorganizations. In contrast, responsible members interact with a subset of the members of the overlay in techniques with a limited scope (C.2). This occurs in the reorganization of clustered overlays, where each cluster has a member acting as a cluster leader interacting with the cluster members and other cluster leaders, while the rest of the members of the cluster only interact with their peers within the cluster. Finally, those techniques where responsible members can make decisions without interacting with other members of the overlay have individual scope (C.3). An example of these techniques is that where sources or intermediate members of the overlay are able to stop media delivery when a certain level of use in their available resources is exceeded. The scope of a technique can be defined algebraically. Let $M_t(n) \subseteq M$ be the members interacting with a responsible member n according to a technique t, then:

$$t \in \text{C.1} \iff M_t(n) = M$$

$$t \in \text{C.2} \iff |M_t(n)| > 1 \wedge M_t(n) \neq M$$

$$t \in \text{C.3} \iff M_t(n) = \emptyset$$

The responsible members for a technique can interact in several ways. The **coordination** between these members and the behavior of each one describe these interactions. All responsible members can act independently, without feedback among them, or can interchange some information. Therefore, self-* techniques can be classified as autonomous (D.1) or collaborative (D.2). The responsible members of an autonomous technique are not required to interact with each other, but they accomplish predefined objectives. On the other hand, collaborative techniques can further be classified as cooperative (D.1.1), when responsible members share efforts to accomplish the same objective, and competitive (D.1.2), when responsible members interact looking for their own benefit. Most of the MCS overlays implement cooperative techniques to achieve self-regulation. These techniques require dynamic adjustments on the sender data bitrate based on feedback from receivers and senders. However, self-regulation can also be achieved using autonomous approaches. Layered media encoding allows for receivers to self-regulate without feedback from any source or other receivers. Finally, self-regulation can also be achieved by competitive approaches. This is the case of those MCSs where overlay members provide more upstream bandwidth to those members from where they are downloading at higher rates.

The **behavior** of responsible members may vary for a technique. This leads to homogeneous (E.1) or heterogeneous (E.2) techniques. All responsible members play similar roles in the former, while different roles of the responsible members can be identified in the latter. Homogeneous techniques are common in fully distributed MCS overlays. In contrast, heterogeneous techniques are based on the client–server model. To formally define behavior, let $b_t(n)$ be the behavior of a responsible member n using a technique t, then:

$$t \in \text{E.1} \iff \forall n, m \in N_t / b_t(n) = b_t(m)$$

$$t \in \text{E.2} \iff \exists n, m \in N_t / b_t(n) \neq b_t(m)$$

Another aspect to consider about a technique is its purpose, referred to as the **strategy** of the technique. Preventive techniques are focused on anticipating, preventing or avoiding an issue (F.1), while corrective techniques react after its occurrence (F.2) alleviating, mitigating or fixing the issue. Most of the MCS use corrective self-organization techniques to optimize, heal or protect the overlay, reorganizing the overlay after some event. Similarly, self-stabilization techniques always require a previous member failure. However, preventive techniques are also possible, such as self-distribution techniques where data is distributed trying to maximize the performance of the overlay and its members.

Self-* techniques in MCSs can also be classified according to the triggering instant. The **triggering** of a technique refers to the instant when the responsible members perform the actions, which leads to synchronous (G.1) and asynchronous (G.2) techniques [61]. The former can be further divided into periodic techniques (G.1.1), when the triggering period is fixed, and aperiodic techniques (G.1.2), when the triggering period is variable. On the other hand, two classes of asynchronous techniques exist: those triggered after an event (G.2.1) and those triggered due to an operational requirement of the overlay (G.2.2). Synchronous techniques are mainly implemented in self-organized overlays by interchanging control messages or taking measurements periodically. However, some real-time communication protocols, such as the Real-time Transport Control Protocol, use a variable period depending on the size of the overlay. The joining and leaving of participants represents one of the events that usually triggers self-organization techniques in MCSs. Both synchronous and asynchronous triggering can be combined.

Finally, some techniques may achieve various self-subproperties simultaneously. The number of the self-subproperties achieved with a technique is referred to as the **cardinality** of the technique. An example of technique with multiple cardinality (H.1) can be found in MCSs self-organized into clusters, as the clustered organization indirectly leads to a self-distributed topology where workload is balanced among the overlay members. Similarly, self-regulation techniques can also lead to self-destruction when some media streams can be eliminated to enhance the quality distribution of others. Otherwise, techniques focused on a single self-subproperty are classified as single cardinality (H.2). Therefore, cardinality can be expressed as follows:

$$\forall t \in T \exists Q' \subseteq Q / t \rightarrow Q'$$

$$t \in \text{H.1} \iff |Q'| > 1$$

$$t \in \text{H.2} \iff |Q'| = 1$$

5. Autonomic techniques in Multimedia Communication Overlays

Autonomic techniques included in MCS overlays fulfilling the aforementioned self-subproperties are analyzed in this section. They are also classified according to the proposed taxonomy.

5.1. Analysis of autonomic techniques

Self-organization techniques are widely implemented in MCSs. They are able to automatically organize their overlays according to configuration, recovery, optimization or preventive objectives. Thus, most of the self-subproperties of MCSs are associated with the capability to self-organize the overlay, since reorganizations of the overlay are usually required to heal or optimize the overlay and prevent future issues. These reorganizations are triggered when detecting issues that may compromise the reliability, availability, integrity or performance of the overlay or periodically to progressively increase these components. Self-organization techniques assess the conditions of the overlays by measuring physical or logical characteristics of the members and their connections. Physical measures assess the performance of member connections using measures such as the communication delay [8,10–14,22,27,34–36,38], available bandwidth [16,18,27,34,35], packet loss rate [18], and member degree [17,24–26,37,38]. Conversely, logical measures assess the usefulness of member connections using data availability and [20,28,30,31,33], upload capacity [23,32], contribution level spreading data across the overlay [15,21,29] or playback locality [19]. Techniques using contribution or upload capacity as measures are inherently competitive. Other non-competitive techniques, such as those based on data availability or playback locality, can be combined and refined with physical measures such as available bandwidth [19,20,28], communication delay [30,31], or member degree [33]. Physical measures can be used to self-organize any kind of MCS, whereas logical measures cannot be used in conferencing MCSs.

Self-organization techniques may help to achieve self-healing subproperties such as self-stabilization. Some self-organization techniques immediately react to stabilize the overlay when member failures occur, reorganizing the overlay to exclude failing members. This can be mainly observed in overlays that are self-organized according to physical measures, especially on conferencing MCSs where a failure highly compromises communications. On the other hand, the self-stabilization of the overlay can be achieved by periodical reorganizations without immediate responses to member failures in self-organized overlays according to logical measures. In this case, the self-organization technique leads to a steady state of the overlay discarding failing members before each reorganization. Self-diagnosis is another self-healing subproperty which is partially related to self-organization. It is focused on detecting failures in the communications between overlay members. Many techniques for detecting failures use timeouts on specific periodical control messages (heartbeats) [16,12,17,23,24,26,29,30,37]. However, the timeouts are associated with self-organization messages in most of the techniques [10,13–15,20–22,25,27,28,31–33,35] rather than using specific control messages. Therefore, self-diagnosis techniques do not rely on self-organization, but sometimes they may take advantage of the self-organization techniques. Other self-diagnosis techniques control the state of the communication channels [8,11,18,19,37]. Self-diagnosis techniques are also implemented for detecting topological conflicts such as the presence of loops [8,11,22,24,31].

Self-organization techniques may be oriented to continuously increase the performance of the overlay, so they help to achieve several self-optimization subproperties. Some MCS overlays use self-organization techniques to initially place members in the overlay, so self-deployment is achieved. In these cases, similar measures to those used when reorganizing the overlay are used to select the initial location of a member joining the overlay. Self-scalability is another self-optimization subproperty that depends on self-organization techniques. A self-scalable overlay must implement a self-organization technique able to cope with the growth of the overlay ensuring optimal connections among overlay members. Furthermore, self-organization techniques can lead to self-distribution. This can be observed in MCSs, mainly on-demand and live streaming MCSs, where self-organization techniques build balanced topologies, such as those clustered-based overlays with a mechanism for preventing oversized or undersized clusters [13,14,25–27,38]. Similarly, there are multitree-based overlays where members self-organize to distribute data using different paths so the forwarding effort is balanced [12,24,31]. These MCSs overlays are also self-contained, since a failure in a link only compromises the path which it belongs. Thus, a failure does not imply excessive degradation in the signal since data will be received through the remaining non-faulty paths. Other criteria used to implement balanced topologies leading to self-distribution are overload, quality of service, upload capacity and priority. In the first case, members serving too many receivers redirect some of the receivers to another less overloaded members [12,22,25,31]. In the second case, members are placed in higher quality streaming trees as their contribution to the delivery process grows [23]. In the third case, members with higher upstream bandwidth are placed closer to data sources [32]. In the last case, members are placed closest to the source according to the number of receivers that they are serving [33].

Similarly, there are also self-protection subproperties such as self-containment and self-immunity that can be achieved with self-organization techniques. Self-containment can emerge from using self-organization techniques where reorganizations are performed according to logical measures, so members are able to identify other members degrading the performance of the overlay. These techniques are usually included in content distribution, on-demand and live streaming MCSs. Commonly used measures are one-to-one performance comparatives [15], down-rate historical scores [21,29,30], or the upload capacity [23,32]. Thus, a self-organization technique can discard useless connections isolating selfish members, so their impact on the overall performance is limited. Self-containment is also achieved relying on self-organization by placing members in the overlay according to unstability [8,33], so the overlay is protected against errors that might occur in these members. This technique is specially suited for tree-based overlays, also leading to self-immune overlays.

On the other hand, there are techniques to fulfill self-optimization subproperties that do not rely on self-organization. For example, some techniques fulfilling self-optimization subproperties such as self-distribution and self-regulation. Two commonly used self-distribution techniques are path scheduling and data scheduling. Path scheduling adapts the network resources to members' demands. A data source is able to share the paths to forward data coming from other sources [14,35]. A member can also vary the number of connections with other members according to their available resources [19]. However, self-distribution is mainly achieved through data scheduling. Information about the availability of data chunks is shared among overlay members so they are able to decide what data should be requested, or retransmitted, and which member should be selected to satisfy each data request. This optimizes the performance of the overlay and also leads to a self-distributed overlay where data is sent according to its priority. There are several approaches to data scheduling that can be observed both in content distribution MCSs and unstructured on-demand and live streaming MCSs. For example, receivers apply data scheduling by requesting first those data chunks with higher playback constraints and fewer replications across the overlay [21], or with a less number of senders [28]. Conversely, senders can apply the following data scheduling strategies: (1) distributing those data chunks which have been requested the least number of times first [29], (2) distributing to the receivers which show the best contributions to the sender [30] or (3) delaying sending some data chunks in favor of others considered more important, where importance is determined according to playback interdependences among data chunks [20,24]. A hierarchy of traffic classes can also be built, assigning a certain percentage of the available network bandwidth to each class [9]. Senders can also collaborate to decide which chunks must be delivered to the receivers by each of them [18].

It should be noted that some of the previously analyzed self-distribution techniques can also be used to achieve self-regulation. For example, data scheduling can be focused on preventing traffic overload on the sender side and service degradation on the receiver side, ensuring a minimum acceptable quality of service [18]. Similarly, path scheduling can be focused on satisfying the streaming quality desired for each member [19]. However, self-regulation is mainly fulfilled by specific techniques such as adaptive streaming, multimedia session negotiation and the use of incentives. Firstly, adaptive streaming adjusts the streaming rate on the fly. In that case, the receivers request the sender to adjust the streaming rate [18,23,36]. In contrast, there are techniques where the sender can detect the degradation of quality according to the feedback from receivers [9,34], or can directly assign the streaming rate to each receiver [34,35]. A self-destructive behavior may also arise from some of these techniques, since the regulation procedure can even stop a media delivery to ensure the reception of streams from another media [35,36]. This can be observed in conferencing MCSs. Another approach to adapt the streaming rate is to deliver data in a hierarchical structure of media layers [20,27,30]. The base layer provides the lowest level of quality while the rest of the layers provide improvements in the quality of the basic content. Layered media also leads to self-containment, since part of the communication can be supported with the basic media layer, so the impact of an error in the enhancement layers is damped, and the degradation of the quality of service is limited. Secondly, multimedia session negotiation configures the service before a member joins the overlay, so the media bitrate is adapted individually [17,37]. Finally, incentive-based techniques allow for modifying the quality of the delivered media according to the overlay member contributions. Thus, overlay members provide more upstream bandwidth to those members from where they are downloading at higher rates [30].

Finally, some self-protection subproperties can also be fulfilled without relying on self-organization. This is the case of self-containment, self-immunity, and self-assessment of risks. Self-containment can be fulfilled using role redundancy, which is based on idle members storing the same information as responsible members [16,17,27], or managing auxiliary communication channels [8,26]. Thus, the overlay is prepared to react when a failure occurs, limiting the impact on the overlay. In the case of self-immunity, secure protocols can be used to enhance data confidentiality, integrity and availability, so the overlay is immune to unauthorized access from untrusted members, or previously banned members [27]. Finally, techniques fulfilling self-assessment of risks are focused on avoiding issues when new members join the overlay, or after data requests. All these techniques try to identify future degradations of the quality of service from the user perspective or the overlay performance. For example, every sender is able to enqueue joining requests when detecting that there is not enough bandwidth in any member of the overlay to incorporate a new member [35]. Alternatively, each member can evaluate the ratio between the upload contribution requested by another member and the benefit of receiving data from such member after a new request, avoiding connections that are not beneficial [32]. Moreover, there are preventive techniques where degradations on the streaming rate between a sender and a receiver are used to predict member failures [18].

5.2. Classification of techniques

Next, the self-* techniques are classified according to the proposed taxonomy.

5.2.1. Self-organization

Table 2 shows the characteristics of the techniques to fulfill self-organization in MCS overlays. As previously commented, all the MCSs include a self-organization technique aimed at healing, optimizing or protecting the overlay. Therefore, self-organization techniques are classified as multiple cardinality as they help to fulfill other self-subproperties.

Most of the techniques are carried out distributedly, sharing the responsibility among all the overlay members. In that case, overlay members usually interact with a subset of the members, so the scope of those techniques is limited. Such interactions are coordinated among overlay members with all the responsible members playing the same role. They are mainly based on taking measurements among overlay members periodically, reacting when member connections improving the performance of the overlay are detected.

There are also several centralized techniques. These techniques are commonly triggered when specific events are identified by a central entity. A client–server model is followed, so the responsibility is shared among the central entity and various key members. This implies that various roles can be identified. There are also centralized techniques where the responsibility is single, so managing decisions are made exclusively by the central entity without interactions with the rest of the overlay.

5.2.2. Self-diagnosis

Table 3 shows the self-diagnosis techniques surveyed. Self-diagnosis techniques are mostly distributed, acting in a corrective

Table 2
Self-organization techniques.

	Respo. (Total, Partial, Single)	Arch. (Centralized, Distributed, Hybrid)	Scope (Global, Limited, Individual)	Coor. (Autonomous, Cooperative, Competitive)	Beh. (Homogeneous, Heterogeneous)	Strat. (Preventive, Corrective)	Trigg. (Periodic, Aperiodic, On Use, On Event)	Card. (Multiple, Single)
According to measurements among members about physical characteristics								
[Jannotti et al. 2000][16]	□■□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Pendarakis et al. 2001][11]	■□□	■□□	□■□	□□■	□■	□■	■□□□	■□
[Roca and El-Sayed 2001][8]	■□□	■□□	□■□	□□■	□■	□■	■□□□	■□
[Banerjee et al. 2002][13]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Chu et al. 2002][10]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Castro et al. 2003][12]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Hefeeda et al. 2003][18]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Luo et al. 2004][35]	■□□	□■□	■□□	□□■	■□	□■	■□□□	■□
[Tran et al. 2004][25]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Liu et al. 2005][34]	□■□	■□□	□■□	□□■	□■	□■	■□□□	■□
[Banerjee et al. 2006][22]	□■□	□■□	□■□	□□■	■□	□■	■□□■	■□
[Song et al. 2006][38]	■□□	□□■	□■□	□□■	□■	□■	■□□■	■□
[Baccichet et al. 2007][24]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Maraviglia et al. 2007][36]	□■□	■□□	□■□	□□■	□■	□■	■□□□	■□
[Zhu et al. 2007][14]	□■□	□□■	□■□	□□■	■□	□■	□□□■	■□
[Yin et al. 2008][27]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Bikfalvi et al. 2010][17]	□□■	■□□	■□□	■□□	■□	□■	□□□■	■□
[Huang et al. 2010][26]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Wang et al. 2010][33]	■□□	□■□	□■□	□□■	□■	□■	■□□□	■□
[Granda et al. 2011][37]	□□■	■□□	■□□	■□□	■□	□■	□□□■	■□
According to measurements among members about logical characteristics								
[Zhang et al. 2005][28]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Hales and Arteconi 2006][15]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Venkataraman et al. 2006][31]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Pianese et al. 2007][29]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Mol et al. 2008][21]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Liu et al. 2009][30]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Ding et al. 2010][20]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Payberah et al. 2010][32]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Qiu et al. 2010][19]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Wang and Chen 2011][23]	■□□	■□□	□■□	□□■	□■	□■	■□□□	■□

manner once failures have occurred. They are triggered periodically and all the members share the responsibility for carrying out the techniques playing similar roles. The main difference between the techniques is the scope. Techniques where members send control messages to all the others, or propagate a detected member failure to the rest of the overlay, are classified as global scope. On the other hand, limited scope techniques are based on members sending control messages to a subset of the overlay.

Another difference among techniques is the triggering instant. In addition to periodical triggering approaches, there are techniques that can be classified as triggered *on use*, since the triggering timeout is based on the data packets or in connection requests derived from the self-organization technique, rather than on specific periodical control messages.

5.2.3. Self-stabilization

Self-stabilization techniques cope with the restoration of communication between overlay members when failures occur. The self-organization technique can reconstruct communications by establishing new connections. As shown in Table 4, there are two main approaches in self-stabilization techniques depending on the relation to self-organization: *on use* or *on event*. In the *on use* approach, self-stabilization is achieved using the overlay ability to self-organize, leading to a steady state when failing members are discarded during the self-organization procedure. Thus, there is no immediate response after a member failure. A member is connected with many, so a failure in one or many members of the overlay should not compromise the performance when using a short interval between reorganizations. In the *on event* approach, members initiate the self-organization procedure after detecting a failure. Most of the techniques analyzed follow this approach.

However, there are overlays where a specific self-stabilization technique differs from the self-organization technique in criteria such as scope, behavior and architecture. One example is global scope self-organization techniques where self-stabilization can be performed re-organizing only a subset of the overlay. Another example is homogeneous self-organization techniques where the actions to stabilize the overlay vary depending on the member that fails.

5.2.4. Self-deployment

Self-deployment techniques manage joining requests from incoming members and their initial placement in the overlay. They mainly implement centralized and distributed architectures as shown in Table 5.

Centralized techniques rely on a single member of the overlay, referred to as the entry member, from which the overlay grows. There are two main trends within this approach. Firstly, the entry member selects the appropriate location (the neighbor member) for the joining member without additional negotiations after the latter contacts the former. Thus, the scope of these techniques is classified as individual. The main difference of techniques within this trend is whether any member can be selected as peer members of the joining member, classified as total-responsibility, or if only a subset of members can be selected, classified as partial-responsibility. Secondly, there are techniques where the entry member provides the joining request with a list of possible neighbor members, and the joining member must determine which are

Table 3
Self-diagnosis techniques.

	Respo. (Total / Partial / Single)	Arch. (Centralized / Distributed / Hybrid)	Scope (Global / Limited / Individual)	Coor. (Autonomous / Cooperative / Competitive)	Beh. (Homogeneous / Heterogeneous)	Strat. (Preventive / Corrective)	Trigg. (Periodic / Aperiodic / On Use / On Event)	Card. (Multiple / Single)
Timeouts over specific control messages								
[Jannotti et al. 2000][16]	□■□	□■□	□■□	□■□	■□	□■	■□□□	□■
[Castro et al. 2003][12]	■□□	□■□	□■□	□■□	■□	□■	■□□□	□■
[Baccichet et al. 2007][24]	■□□	□■□	□■□	□■□	■□	□■	■□□□	□■
[Pianese et al. 2007][29]	■□□	□■□	□■□	□■□	■□	□■	■□□□	□■
[Liu et al. 2009][30]	■□□	□■□	□■□	□■□	■□	□■	□■□□	□■
[Bikfalvi et al. 2010][17]	■□□	□■□	□■□	□■□	■□	□■	□■□□	□■
[Huang et al. 2010][26]	■□□	□■□	□■□	□■□	■□	□■	■□□□	□■
[Granda et al. 2011][37]	□■□	□■□	□■□	□■□	■□	□■	□■□□	□■
[Wang and Chen 2011][23]	■□□	□■□	□■□	□■□	■□	□■	■□□□	□■
Timeouts over self-organization messages								
[Banerjee et al. 2002][13]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Chu et al. 2002][10]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Luo et al. 2004][35]	■□□	□■□	■□□	□■□	■□	□■	□□■□	□■
[Tran et al. 2004][25]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Zhang et al. 2005][28]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Banerjee et al. 2006][22]	□■□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Hales and Arteconi 2006][15]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Venkataraman et al. 2006][31]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Zhu et al. 2007][14]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Mol et al. 2008][21]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Yin et al. 2008][27]	□■□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Ding et al. 2010][20]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Payberah et al. 2010][32]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Wang et al. 2010][33]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
Controlling communications channels								
[Pendarakis et al. 2001][11]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Roca and El-Sayed 2001][8]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Hefeeda et al. 2003][18]	■□□	□■□	□■□	■□□	■□	□■	□□■□	□■
[Qiu et al. 2010][19]	■□□	□■□	□■□	■□□	■□	□■	□□■□	□■
[Granda et al. 2011][37]	□□■	■□□	□■□	■□□	■□	□■	□□■□	□■
Detecting structural issues derived from a specific topology								
[Pendarakis et al. 2001][11]	■□□	■□□	■□□	□■□	□■	□■	□□■□	□■
[Roca and El-Sayed 2001][8]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Banerjee et al. 2006][22]	□■□	□■□	□■□	■□□	■□	□■	□□□■	□■
[Venkataraman et al. 2006][31]	■□□	□■□	□■□	□■□	■□	□■	□□■□	□■
[Baccichet et al. 2007][24]	■□□	□■□	□■□	□■□	■□	□■	□□□■	□■

the most appropriate. These techniques are classified as limited scope, since some kind of negotiation is needed to join the overlay. They can also be divided into total-responsibility techniques when any member of the overlay can be selected as a candidate neighbor by the entry member, and partial-responsibility if only certain members can accommodate new joins.

On the other hand, there are self-deployment techniques based on a distributed architecture. Most of them are classified as total-responsibility since each member of the overlay can play the role of an entry member. These techniques are also classified as individual scope, since no more interactions are needed after localizing an entry member.

5.2.5. Self-scalability

Self-scalable techniques should provide mechanisms to ensure performance while facing an increasing demand. High performance in MCS overlays is equivalent to guaranteeing a minimum quality of service, low latency or a compromise between the two as the workload of the overlay increases. MCS overlays with strict latency constraints are inherently limited in scalability, since latency usually increases with the size of the overlay. Similarly, MCS overlays with a centralized self-organization technique are also hardly scalable. Self-scalability also depends on how the overlay topology is managed. Thus, mesh-based overlays in which a member must send control messages to the whole overlay are difficult to scale. Furthermore, it is not possible to ensure scaling from a certain level of width (a member could serve many receivers) or from a certain level of depth (the latency is increased in each level) in tree-based overlays, although this can be mitigated using clustering approaches. Therefore, those overlays that are not latency-dependent, where a distributedly self-organization technique without global interaction among all the members of the overlay is used, can be considered as self-scalable. Table 6 summarizes these self-organization techniques.

5.2.6. Self-distribution

Table 7 shows the self-distribution techniques observed in MCS overlays. All of them are distributed, and the responsible members interact with a subset of the overlay rather than with the whole overlay. The responsibility to carry out the techniques is always shared among overlay members, globally shared in some techniques and partially shared in others. Most of the path scheduling techniques are triggered when a certain network or member issue occurs, while data scheduling techniques are usually triggered

Table 4
Self-stabilization techniques.

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Based on their self-organization technique								
[Jannotti et al. 2000][16]	□■□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Roca and El-Sayed 2001][8]	■□□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Banerjee et al. 2002][13]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Chu et al. 2002][10]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Castro et al. 2003][12]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Hefeeda et al. 2003][18]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Luo et al. 2004][35]	■□□	□■□	■□□	□□■	■□	□■	□□□■	□■
[Tran et al. 2004][25]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Zhang et al. 2005][28]	■□□	□■□	□■□	□□■	■□	□■	□□■□	□■
[Banerjee et al. 2006][22]	□■□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Hales and Arteconi 2006][15]	■□□	□■□	□■□	□□■	■□	□■	□□■□	□■
[Baccichet et al. 2007][24]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Pianese et al. 2007][29]	■□□	□■□	□■□	□□■	■□	□■	□□■□	□■
[Zhu et al. 2007][14]	□■□	□□■	□■□	□□■	■□	□■	□□□■	□■
[Mol et al. 2008][21]	■□□	□■□	□■□	□□■	■□	□■	□□■□	□■
[Yin et al. 2008][27]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Liu et al. 2009][30]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Bikfalvi et al. 2010][17]	□□■	■□□	■□□	■□□	■□	□■	□□□■	□■
[Ding et al. 2010][20]	■□□	□■□	□■□	□□■	■□	□■	□□■□	□■
[Huang et al. 2010][26]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Payberah et al. 2010][32]	■□□	□■□	□■□	□□■	■□	□■	□□■□	□■
[Qiu et al. 2010][19]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Wang et al. 2010][33]	■□□	□■□	□■□	□□■	□■	□■	□□□■	□■
[Granda et al. 2011][37]	□□■	■□□	■□□	■□□	■□	□■	□□□■	□■
[Wang and Chen 2011[23]	■□□	■□□	□■□	□□■	□■	□■	□□□■	□■
Different from their self-organization technique								
[Pendarakis et al. 2001][11]	■□□	□■□	□■□	□□■	■□	□■	□□□■	□■
[Venkataraman et al. 2006][31]	■□□	□■□	□■□	□■□	□■	□■	□□□■	□■

Table 5
Self-deployment techniques.

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Managing joining requests								
[Jannotti et al. 2000][16]	□■□	□□■	□■□	□□■	□■	□■	□□□■	□■
[Pendarakis et al. 2001][11]	■□□	■□□	□□■	□■□	□■	□■	□□□■	□■
[Banerjee et al. 2002][13]	■□□	□□■	□■□	□□■	□■	□■	□□□■	□■
[Chu et al. 2002][10]	■□□	□■□	□□■	□■□	■□	□■	□□□■	□■
[Castro et al. 2003][12]	■□□	□□■	□■□	□■□	□■	□■	□□□■	□■
[Hefeeda et al. 2003][18]	■□□	□■□	□□■	□■□	■□	□■	□□□■	□■
[Luo et al. 2004][35]	■□□	□■□	□□■	□■□	■□	□■	□□□■	□■
[Tran et al. 2004][25]	■□□	□□■	□■□	□□■	□■	□■	□□□■	□■
[Liu et al. 2005][34]	□■□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Zhang et al. 2005][28]	■□□	■□□	□□■	□■□	□■	□■	□□□■	□■
[Banerjee et al. 2006][22]	□■□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Venkataraman et al. 2006][31]	■□□	□■□	□□■	□■□	■□	□■	□□□■	□■
[Baccichet et al. 2007][24]	■□□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Pianese et al. 2007][29]	■□□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Zhu et al. 2007][14]	□■□	□■□	□□■	□■□	■□	□■	□□□■	□■
[Yin et al. 2008][27]	□■□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Liu et al. 2009][30]	■□□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Bikfalvi et al. 2010][17]	■□□	■□□	□□■	□■□	□■	□■	□□□■	□■
[Ding et al. 2010][20]	■□□	■□□	□□■	□■□	□■	□■	□□□■	□■
[Huang et al. 2010][26]	■□□	■□□	□□■	□■□	□■	□■	□□□■	□■
[Payberah et al. 2010][32]	□■□	■□□	□□■	□■□	□■	□■	□□□■	□■
[Qiu et al. 2010][19]	■□□	■□□	□■□	□■□	□■	□■	□□□■	□■
[Wang et al. 2010][33]	■□□	■□□	□■□	□□■	□■	□■	□□□■	□■
[Granda et al. 2011][37]	□■□	■□□	□□■	□■□	□■	□■	□□□■	□■
[Wang and Chen 2011][23]	■□□	■□□	□■□	□□■	□■	□■	□□□■	□■

Table 6
Self-scalability techniques.

	Resp. (Total / Partial / Single)	Arch. (Centralized / Distributed / Hybrid)	Scope (Global / Limited / Individual)	Coor. (Autonomous / Cooperative / Competitive)	Beh. (Homogeneous / Heterogeneous)	Strat. (Preventive / Corrective)	Trigg. (Periodic / Aperiodic / On Use / On Event)	Card. (Multiple / Single)
Self-organization techniques from which self-scaling emerges								
[Banerjee et al. 2002][13]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Chu et al. 2002][10]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Castro et al. 2003][12]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Hefeeda et al. 2003][18]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Tran et al. 2004][25]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Zhang et al. 2005][28]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Banerjee et al. 2006][22]	□■□	□■□	□■□	□□■	■□	□■	■□□■	■□
[Venkataraman et al. 2006][31]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Baccichet et al. 2007][24]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Pianese et al. 2007][29]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Zhu et al. 2007][14]	□■□	□□■	□■□	□□■	■□	□■	□□□■	■□
[Yin et al. 2008][27]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Liu et al. 2009][30]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Ding et al. 2010][20]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Huang et al. 2010][26]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Payberah et al. 2010][32]	■□□	□■□	□■□	□□■	■□	□■	■□□□	■□
[Qiu et al. 2010][19]	■□□	□■□	□■□	□□■	■□	□■	□□□■	■□
[Wang et al. 2010][33]	■□□	□■□	□■□	□□■	□■	□■	■□□□	■□

Table 7
Self-distribution techniques.

	Respo. (Total / Partial / Single)	Arch. (Centralized / Distributed / Hybrid)	Scope (Global / Limited / Individual)	Coor. (Autonomous / Cooperative / Competitive)	Beh. (Homogeneous / Heterogeneous)	Strat. (Preventive / Corrective)	Trigg. (Periodic / Aperiodic / On Use / On Event)	Card. (Multiple / Single)
Path scheduling								
[Luo et al. 2004][35]	■□□	□■□	□■□	□■□	■□	□■	□□□■	□■
[Zhu et al. 2007][14]	□■□	□■□	□■□	□■□	■□	□■	□□□■	□■
[Qiu et al. 2010][19]	■□□	□■□	□■□	□■□	■□	■□	□□□■	■□
Data scheduling								
[Chawathe et al. 2000][9]	□■□	□■□	□□■	■□□	■□	■□	□□□■	□■
[Hefeeda et al. 2003][18]	■□□	□■□	□■□	□■□	■□	■□	□□□■	■□
[Zhang et al. 2005][28]	■□□	□■□	□■□	□■□	■□	■□	□□□■	□■
[Baccichet et al. 2007][24]	■□□	□■□	□□■	■□□	■□	■□	□□■□	□■
[Pianese et al. 2007][29]	■□□	□■□	□□■	■□□	■□	■□	□□□■	□■
[Mol et al. 2008][21]	■□□	□■□	□■□	□■□	■□	■□	□□□■	□■
[Liu et al. 2009][30]	■□□	□■□	□■□	□□■	■□	■□	□□□■	□■
[Ding et al. 2010][20]	■□□	□■□	□□■	■□□	■□	■□	□□■□	□■

before distributing data. Thus, the former follow a corrective strategy while the latter follow a preventive strategy. Different actions are performed by senders and receivers in many self-distribution techniques, so the techniques are classified as heterogeneous. Some of the presented techniques are focused on ensuring a minimum acceptable quality of service also promoting self-regulation, so they are classified as multiple cardinality.

5.2.7. Self-regulation

Table 8 summarizes the self-regulation techniques surveyed. As can be seen, there is a wide variety according to responsibility approaches. Some of the techniques are exclusively managed by the data sources, while the data sources collaborate with intermediary entities in other techniques. There are also techniques implying a negotiation among senders and receivers across the whole overlay. Thus, the responsibility relies on one, several or the whole overlay respectively. Similarly to self-distribution techniques, many of the self-regulation techniques follow a client–server model, so the role played by each responsible member may vary. In addition, most of the techniques are triggered preventively when particular circumstances are identified, or according to an *on use* approach. This *on use* approach is based on manipulating the data distributed throughout the overlay to anticipate issues related to quality of service at the receivers. Furthermore, some of these techniques have multiple cardinality as they may also lead to self-destruction, by adaptive streaming techniques that preserve the quality of a media stopping another one, or self-containment using layered media.

5.2.8. Self-assessment of risks

As can be seen in Table 9, techniques to fulfill self-assessment of risks are quite similar. All of them follow a preventive approach, trying to anticipate and avoid threatening issues. They also follow a distributed architecture, sharing the responsibility among all the overlay members. The main differences between them arise from how they are triggered. Some of them are triggered when specific

Table 8
Self-regulation techniques.

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Adaptive streaming								
[Chawathe et al. 2000][9]	□■□	□■□	□■□	□■□	□■	□■	□□□■	□■
[Hefeeda et al. 2003[18]	■□□	□■□	□■□	□■□	□■	□■	□□□■	□■
[Luo et al. 2004][35]	■□□	□■□	■□□	□■□	■□	■□	□□□■	■□
[Liu et al. 2005][34]	■□□	□□■	□■□	□■□	□■	□■	□□□■	□■
[Maraviglia et al. 2007][36]	■□□	□■□	□■□	□■□	□■	□■	□□□■	■□
[Wang and Chen 2011][23]	■□□	□■□	□■□	□■□	□■	□■	□□□■	□■
Layered video distribution								
[Yin et al. 2008][27]	■□□	□■□	□■□	□■□	□■	■□	□□■□	■□
[Liu et al. 2009][30]	■□□	□■□	□■□	□■□	□■	■□	□□■□	■□
[Ding et al. 2010][20]	■□□	□■□	□■□	□■□	□■	■□	□□■□	■□
Incentive strategies								
[Liu et al. 2009][30]	■□□	□■□	□■□	□□■	■□	■□	□□■□	□■
Multimedia session negotiation								
[Bikfalvi et al. 2010][17]	■□□	■□□	□■□	□■□	□■	■□	□□□■	□■
[Granda et al. 2011][37]	■□□	■□□	□■□	□■□	□■	■□	□□□■	□■

Table 9
Techniques for self-assessment of risks.

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Controlling degradation of service								
[Hefeeda et al. 2003][18]	■□□	□■□	□■□	□■□	■□	■□	□□■□	□■
[Luo et al. 2004][35]	■□□	□■□	■□□	□■□	■□	■□	□□□■	□■
[Payberah et al. 2010][32]	■□□	□■□	□■□	□□■	□■	■□	□□□■	□■

threats are detected, while other techniques are triggered *on use*, assessing data distribution.

5.2.9. Self-containment and self-immunity

Tables 10 and 11 depict the characteristics of the self-containment and self-immunity techniques respectively. There are mainly two approaches in the first two groups of self-containment techniques: distributed techniques according to global responsibility where each member manages a subset of the overlay, or centralized techniques where a central entity carries out the technique, managing the whole overlay. In contrast, role redundancy techniques are quite similar in their approaches. However, it turns out that although self-immunity is closely related to self-containment, the former seems to be less present in MCS overlay management. In fact, most of the self-containment techniques do not provide self-immunity capabilities. Only the techniques focused on keeping away potentially faulty members are classified as multiple cardinality since they lead to a certain degree of self-immunization. Those unstable members are moved to a non-compromising overlay region, so the failure in one of these members does not affect the rest of the overlay.

The reason for this weak relation between self-containment and self-immunity is that self-immunity is more difficult to accomplish than self-containment in certain MCSs such as open P2P systems. This difficulty can be illustrated by building an immune MCS overlay facing selfish behaviors. Such an overlay could share the historical behavior of each member, and thus identify selfish behaviors. This can work well in an overlay with authenticated access. However, there are several threats that invalidate this approach in open MCS overlays, whereas self-containment performs successfully to cope with selfish members. For instance, a member with the ability to adopt several identities using the same selfish strategy cannot be detected and subsequently isolated. Thus, a continuous analysis of the overlay members' behavior would be needed to prevent these malicious actions, making an effective immunization procedure impossible. Therefore, although the detection and isolation of a selfish member can be assumed, other containment actions are required to avoid such a member compromising the overlay performance again.

5.2.10. Self-destruction and self-installation

Table 11 also shows the characteristics of self-destruction and self-installation techniques. It should be noted that the implementation of both self-subproperties is limited in MCS overlays.

A preventive self-destruction technique is included in the data source in [29]. Data source lacks a data feedback mechanism, which could be exploited by selfish members to retrieve data directly from the source without contributing to the MCS. Therefore, the source periodically changes the subset of served members to mitigate this threat.

A self-installation technique can be found in [15]. Members are able to adapt their behavior by copying the contribution strategies from their neighbors. The authors point that this technique could be achieved using mobile code, which allows overlay members to

Table 10
Self-containment techniques.

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Selfish prevention								
[Hales and Arteconi 2006][15]	■□□	□■□	□■□	□□■	■□	■□	■□□□	□■
[Pianese et al. 2007][29]	■□□	□■□	□■□	□□■	■□	□■	□□■□	□■
[Mol et al. 2008][21]	■□□	□■□	□■□	□□■	■□	□■	■□□□	□■
[Liu et al. 2009][30]	■□□	□■□	□■□	□□■	■□	□■	□□■□	■□
[Wang and Chen 2011][23]	■□□	■□□	□■□	□□■	□■	□■	□□■□	□■
Unstability prevention								
[Roca and El-Sayed 2001][8]	□□■	■□□	■□□	■□□	■□	■□	■□□□	■□
[Wang et al. 2010][33]	■□□	□■□	□□■	■□□	■□	■□	■□□□	■□
Role redundancy								
[Jannotti et al. 2000][16]	□■□	□■□	□■□	□■□	■□	■□	□□□■	□■
[Roca and El-Sayed 2001][8]	□□■	■□□	■□□	■□□	■□	■□	□□□■	□■
[Yin et al. 2008][27]	□■□	□■□	□■□	□■□	■□	■□	■□□□	□■
[Bikfalvi et al. 2010][17]	□■□	■□□	□■□	□■□	□■	■□	□□□■	□■
[Huang et al. 2010][26]	□■□	□■□	□■□	□■□	■□	■□	□□□■	□■

Table 11
Self-immunity, self-destruction and self-installation techniques.

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Facing unauthorized access								
[Yin et al. 2008][27]	□■□	□□■	□■□	□■□	□■	■□	□□□■	□■

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Continuous changing the set of receivers								
[Pianese et al. 2007][29]	□□■	■□□	■□□	■□□	■□	■□	■□□□	□■

	Respo.	Arch.	Scope	Coor.	Beh.	Strat.	Trigg.	Card.
	Total Partial Single	Centralized Distributed Hybrid	Global Limited Individual	Autonomous Cooperative Competitive	Homogeneous Heterogeneous	Preventive Corrective	Periodic Aperiodic On Use On Event	Multiple Single
Copying neighbor's contribution strategies								
[Hales and Arteconi 2006][15]	■□□	□■□	□■□	□■□	■□	□■	■□□□	□■

execute previously unavailable software. This technique is based on periodical one-to-one comparatives where two members evaluate their contribution strategy. The responsibility to carry out this technique relies on all the overlay leading to a fully-distributed architecture. The evaluation results in one member copying the strategy of the other.

6. Conclusions

The complexity of MCS overlays has led to many MCSs achieving self-management characteristics. Therefore, autonomic properties such as self-organization, self-deployment, self-diagnosis and self-stabilization are common in many MCS overlay management techniques. Other self-subproperties such as self-regulation and self-distribution have also been achieved, but they have less presence. The growth and improvement of MCSs will depend on the constant evolution and wider adoption of autonomic computing to manage their overlays.

The analysis of self-properties in overlay management reveals that self-protection is the least covered. However, there are interesting self-protection subproperties, such as self-containment and self-immunity, that are research fields with still open issues, so new techniques may arise to achieve them.

MCS overlay management only includes some of the analyzed self-subproperties. Therefore, a more exhaustive translation of the autonomic computing paradigm is still needed, since none of the analyzed MCSs fulfills the basic fundamentals of autonomic computing properly.

The principal contribution of this paper is twofold: presenting the different self-subproperties that can be found in MCS overlay management, and proposing a taxonomy to classify and compare the techniques to achieve such self-subproperties. This can be used as a unified framework to analyze and compare MCS overlays.

Future work focuses on the analysis of new techniques or approaches to fulfill self-subproperties in MCS overlays. The proposed taxonomy may also be adapted to cope with the challenges imposed by new requirements and implementations of MCSs.

Acknowledgments

This research has been partially funded by the Spanish National Plan of Research, Development and Innovation under the project TIN2008-06045-C02. Financial support (Grant:UNOV-10-BECDOC-S) given by the University of Oviedo-Banco de Santander is also acknowledged.

References

[1] P. Horn, Autonomic computing: IBM's perspective on the state of information technology, IBM's Autonomic Computing Manifesto (2001) 1–40.
[2] J.O. Kephart, D.M. Chess, The vision of autonomic computing, Computer 36 (2003) 41–50.
[3] A.G. Ganek, T.A. Corbi, The dawning of the autonomic computing era, IBM Systems Journal 42 (1) (2003) 5–18.
[4] R. Sterritt, D.W. Bustard, Autonomic computing – a means of achieving dependability? in: Proc. 10th IEEE Int. Conf. on Engineering of Computer-Based Systems (ECBS 2003), Huntsville, AL, USA, 2003, pp. 247–251.
[5] R. Sterritt, Autonomic computing, Innovations in Systems and Software Engineering 1 (1) (2005) 79–88.
[6] C. Yeo, B. Lee, M. Er, A survey of application level multicast techniques, Computer Communications 27 (15) (2004) 1547–1568.
[7] A. El-Sayed, V. Roca, L. Mathy, A survey of proposals for an alternative group communication service, IEEE Network 17 (1) (2003) 46–51.
[8] V. Roca, A. El-Sayed, Host-based multicast (HBM) solution for group communications, in: Proc. First IEEE Int. Conf. on Networking (ICN 2001), vol. 2093, Colmar, France, 2001, pp. 610–619.
[9] Y. Chawathe, S. McCanne, E.A. Brewer, RMX: reliable multicast for heterogeneous networks, in: Proc 19th IEEE Int. Conf. on Computer Communications (INFOCOM 2000), vol. 2, Tel-Aviv, Israel, 2000, pp. 795–804.
[10] Y. Chu, S.G. Rao, S. Seshan, H. Zhang, A case for end system multicast, IEEE Journal on Selected Areas in Communications 20 (8) (2002) 1456–1471.
[11] D. Pendarakis, S. Shi, D. Verma, M. Waldvogel, ALMI: an application level multicast infrastructure, in: Proc. Third USENIX Symp. on Internet Technologies and Systems (USITS 2001), San Francisco, CA, USA, 2001, pp. 49–60.
[12] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron, A. Singh, SplitStream: high-bandwidth multicast in cooperative environments, ACM Operating Systems Review 37 (5) (2003) 298–313.
[13] S. Banerjee, B. Bhattacharjee, C. Kommareddy, Scalable application layer multicast, in: Proc. 17th ACM SIGCOMM Int. Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2002), Pittsburgh, PA, USA, 2002, pp. 205–217.
[14] J. Zhu, T. Tsuchiya, K. Koyanagi, Peer-to-peer collaborative application-level multicast, in: Proc. Ninth IEEE Int. Conf. on E-Commerce Technology; Fourth IEEE Int. Conf. on Enterprise Computing, E-Commerce and E-Services (CEC/EEE 2007), Tokyo, Japan, 2007, pp. 228–235.
[15] D. Hales, S. Arteconi, SLACER: a self-organizing protocol for coordination in peer-to-peer networks, IEEE Intelligent Systems 21 (2) (2006) 29–35.
[16] J. Jannotti, D.K. Gifford, K.L. Johnson, M.F. Kaashoek, J.W. O'Toole Jr., Overcast: reliable multicasting with on overlay network, in: Proc. Fourth USENIX Int. Conf. and Symp. on Operating System Design & Implementation (OSDI '00), Berkeley, CA, USA, 2000, pp. 14–29.
[17] A. Bikfalvi, J. García-Reinoso, I. Vidal, F. Valera, A peer-to-peer IPTV service architecture for the IP multimedia subsystem, International Journal of Communication Systems 23 (6–7) (2010) 780–801.
[18] M.M. Hefeeda, A. Habib, B. Botev, D. Xu, B.K. Bhargava, PROMISE: peer-to-peer media streaming using collectcast, in: Proc. 11th ACM Int. Conf. on Multimedia (MM '03), Berkeley, CA, USA, 2003, pp. 45–54.
[19] X. Qiu, W. Huang, C. Wu, F. Lau, X. Lin, InstantLeap: an architecture for fast neighbor discovery in large-scale P2P VoD streaming, Multimedia Systems 16 (3) (2010) 183–198.
[20] Y. Ding, J. Liu, D. Wang, H. Jiang, Peer-to-peer video-on-demand with scalable video coding, Computer Communications 33 (14) (2010) 1589–1597.
[21] J.J.D. Mol, J.A. Pouwelse, M. Meulpolder, D.H.J. Epema, 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), vol. 6818, San Jose, CA, USA, 2008, pp. 681804–681804-8.
[22] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, S. Khuller, OMNI: an efficient overlay multicast infrastructure for real-time applications, Computer Networks 50 (6) (2006) 826–841.
[23] W. Wang, Y. Chen, SmartPeerCast: a smart QoS driven P2P live streaming framework, Multimedia Tools and Applications 54 (2) (2011) 445–471.
[24] P. Baccichet, J. Noh, E. Setton, B. Girod, Content-aware P2P video streaming with low latency, in: Proc. IEEE Int. Conf. on Multimedia and Expo (ICME 2007), Beijing, China, 2007, pp. 400–403.
[25] D.A. Tran, K. Hua, S. Member, T. Do, A peer-to-peer architecture for media streaming, IEEE Journal on Selected Areas in Communications 22 (1) (2004) 121–133.
[26] N.-F. Huang, Y.-J. Tzang, H.-Y. Chang, C.-W. Ho, Enhancing P2P overlay network architecture for live multimedia streaming, Information Sciences 180 (17) (2010) 3210–3231.
[27] H. Yin, C. Lin, Q. Zhang, Z. Chen, D. Wu, TrustStream: a secure and scalable architecture for large-scale internet media streaming, IEEE Transactions on Circuits and Systems for Video Technology 18 (12) (2008) 1692–1702.
[28] X. Zhang, J. Liu, B. Li, T. Shing Peter 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), vol. 3, Miami, FL, USA, 2005, pp. 2102–2111.
[29] F. Pianese, D. Perino, J. Keller, E. Biersack, PULSE: an adaptive, incentive-based, unstructured P2P live streaming system, IEEE Transactions on Multimedia 9 (8) (2007) 1645–1660.
[30] Z. Liu, Y. Shen, K.W. Ross, S.S. Panwar, Y.W.Y., LayerP2P: using layered video chunks in P2P live streaming, IEEE Transactions on Multimedia 11 (7) (2009) 1340–1352.
[31] V. Venkataraman, K. Yoshida, P. Francis, Chunkyspread: heterogeneous unstructured tree-based peer-to-peer multicast, in: Proc. 14th IEEE Int. Conf. on Network Protocols (ICNP 2006), Santa Barbara, CA, USA, 2006, pp. 2–11.
[32] A. Payberah, J. Dowling, F. Rahimian, S. Haridi, GradienTv: 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), LNCS, vol. 6115, 2010, pp. 212–225.
[33] F. Wang, Y. Xiong, J. Liu, MTreebone: a collaborative tree-mesh overlay network for multicast video streaming, IEEE Transactions on Parallel and Distributed Systems 21 (3) (2010) 379–392.
[34] Z. Liu, G. He, 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), Wuham, China, 2005, pp. 1265–1270.
[35] C. Luo, J. Li, S. Li, DigiMetro: an application-level multicast system for multi-party video conferencing, in: Proc. IEEE Int. Conf. on Global Telecommunications (GLOBECOM '04), vol. 2, Dallas, TX, USA, 2004, pp. 982–987.
[36] G. Maraviglia, M. Masi, V. Merlo, F. Licandro, A. Russo, 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), Patras, Greece, 2007, pp. 1172–1179.
[37] J. Granda, P. Nuño, D. García, F. Suárez, Towards resilient synchronous e-training platforms, in: Proc 10th IEEE Int. Symp. on Network Computing and Applications (NCA 2011), Cambridge, MA, USA, 2011, pp. 251–256.
[38] H. Song, D.S. Lee, H.R. Oh, Application layer multicast tree for real-time media delivery, Computer Communications 29 (9) (2006) 1480–1491.
[39] A. Berns, S. Ghosh, Dissecting self-* properties, in: Proc. Third IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems (SASO 2009), Los Alamitos, CA, USA, 2009, pp. 10–19.
[40] M. Mamei, R. Menezes, R. Tolksdorf, F. Zambonelli, Case studies for self-organization in computer science, Journal of Systems Architecture 52 (2006) 443–460.
[41] A. Haydarlou, B. Overeinder, M. Oey, F. 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) LNCS, vol. 4158, 2006, pp. 67–77.
[42] M.W. Shapiro, Self-healing in modern operating systems, Queue 2 (2004) 66–75.
[43] S. Dolev, Self-Stabilization, MIT Press, Cambridge, MA, USA, 2000.
[44] S. Bouchenak, F. Boyer, D. Hagimont, S. Krakowiak, A. Mos, N. de Palma, V. Quema, J.-B. Stefani, Architecture-based autonomous repair management: an application to J2EE clusters, in: Proc. 24th IEEE Symposium on Reliable Distributed Systems (SRDS 2005), Orlando, FL, USA, 2005, pp. 13–24.
[45] A. Howard, M.J. Matarić, Gaurav S. Sukhatme, An incremental self-deployment algorithm for mobile sensor networks, Autonomous Robots 13 (2002) 113–126.
[46] P. Flocchini, G. Prencipe, N. Santoro, Self-deployment of mobile sensors on a ring, Theoretical Computer Science 402 (1) (2008) 67–80.
[47] P.A. Felber, E. 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) LNCS, vol. 3460, 2005, pp. 343–357.
[48] M.J. Oudshoorn, M.M. Fuad, D. Deb, Towards autonomic computing: injecting self-organizing and self-healing properties into java programs, in: Proc. Fifth Int. Conf. on New Trends in Software Methodologies, Tools and Techniques (SOMET 06), Amsterdam, The Netherlands, 2006, pp. 384–406.

[49] P.A. Felber, E. 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) LNCS, vol. 3460, 2005, pp. 343–357.

[50] M.J. Oudshoorn, M.M. Fuad, D. Deb, Towards autonomic computing: injecting self-organizing and self-healing properties into java programs, in: Proc. Fifth Int. Conf. on New Trends in Software Methodologies, Tools and Techniques (SOMET 06), Amsterdam, The Netherlands, 2006, pp. 384–406.

[51] J. Molleda, R. Usamentiaga, D. Garcia, F. 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, vol. 7877, San Francisco, CA, USA, 2011, pp. 1–13.

[52] B. Li, 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) (1999) 1632–1650.

[53] E. Damiani, D.C. di Vimercati, S. Paraboschi, P. Samarati, F. Violante, A reputation-based approach for choosing reliable resources in peer-to-peer networks, in: Proc. Ninth ACM Int. Conf. on Computer and Communications Security (CCS '02), New York, NY, USA, 2002, pp. 207–216.

[54] R. Sterritt, M. Hinchey, SPAACE IV: self-properties for an autonomous & autonomic computing environment – Part IV A Newish hope, in: Proc. Seventh IEEE Int. Conf. and Workshops on Engineering of Autonomic and Autonomous Systems (EASE '10), Washington, DC, USA, 2010, pp. 119–125.

[55] L. Beaudoin, N. Japkowicz, S. Matwin, Autonomic computer network defence using risk state and reinforcement learning, in: Proc. First Int. Conf. on Cyber Warfare (CCD COE), No. MR61274, Tallinn, Estonia, 2009.

[56] F. Chiang, J. Agbinya, R. Braun, Risk and vulnerability assessment of secure autonomic communication networks, in: Proc. Second Int. Conf. on Wireless Broadband and Ultra Wideband Communications (AUSWIRELESS '07), Washington, DC, USA, 2007, pp. 40–46.

[57] A. Mahajan, F. Figueroa, An automatic self-installation and calibration method for a 3D position sensing system using ultrasonics, Robotics and Autonomous Systems 28 (4) (1999) 281–294.

[58] H. Tianfield, R. Unland, Towards autonomic computing systems, Engineering Applications of Artificial Intelligence 17 (7) (2004) 689–699.

[59] M. Rahman, R. Ranjan, R. Buyya, A taxonomy of autonomic application management in grids, in: Proc. 16th IEEE Int. Conf. on Parallel and Distributed Systems (ICPADS 2010), Shanghai, China, 2010, pp. 189 –196.

[60] T. De Wolf, T. Holvoet, A taxonomy for self-properties in decentralised autonomic computing, Autonomic Computing: Concepts, Infrastructure, and Applications (2006) 101–120.

[61] L. Alima, S. Haridi, A. Ghodsi, S. El-Ansary, P. Brand, Self-properties in distributed k-ary structured overlay networks., in: Proc. First Int. Workshop on Self-* Properties in Complex Information Systems (SELF-* 2004), Bertinoro, Forlí-Cesena, Italy, 2004.

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.

2010 Fifth International Conference on Systems and Networks Communications

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

Pelayo Nuño, Juan C. Granda, Daniel F. García, and Francisco J. Suárez
Computer Department
University of Oviedo
Gijón, Spain
e-mail: {nunopelayo,jcgranda,dfgarcia,fjsuarez}@uniovi.es

***Abstract*—Synchronous e-learning platforms are frequently used to train the personnel of large corporations. Usually, these corporations are geographically dispersed in multiple sites, so efficient transport of multimedia streams during an e-learning activity is essential. In order to interconnect the groups of participants, it is necessary to deploy a communication mesh. If the number of sites is high, non-automatic deployment of the communication mesh is complex, cumbersome and prone to errors, delaying the startup of e-learning activities. In this paper, an automatic deployment technique based on a centralized server is proposed. This technique continuously updates the composition of the communication mesh, and involves only those sites where there are active participants in the e-learning activity, modifying the mesh topology in function of the joining and leaving of participants. Tests show that the proposed automatic deployment technique makes use of the available bandwidth efficiently.**

Keywords - automatic deployment; communication mesh; synchronous e-learning; rendezvouz point; multicast emulation.

I. INTRODUCTION

Research into the transmission of multimedia streams over the Internet has grown considerably in recent years. As a result, new services based on real-time communication have been developed. One example of such services is synchronous e-learning [1]. It is common for dispersed groups of users to participate in synchronous e-learning activities. All the multimedia streams generated during an activity must be forwarded to each of the participants ensuring low transmission delay, minimal bandwidth consumption, and low computational requirements on the end systems. Synchronous e-learning platforms are typically used to develop training programs for human resources in large corporations.

There are several techniques to provide communication between dispersed groups of participants. Those that emulate the behavior of multicast networks are particularly useful, as the multicast communication ensures that the number of streams interchanged between participants is minimal, providing the synchronous e-learning systems with the maximum level of scalability. El-Sayed et al. [2] reviewed the techniques that emulate multicast networks, grouping them into the following classes: unicast/multicast reflectors, application-level multicast (ALM) and peer-to-peer architectures (P2P). All these techniques can support e-learning activities, building a communication mesh oriented to the transport of multimedia streams. A common challenge for all these techniques is to provide easy and automatic deployment of the communication mesh between all the groups.

These basic techniques can also be combined. An example of combination is the technique developed by Granda et al. [3] to communicate dispersed groups of participants in an e-learning activity. Each group is located in a site of a large, geographically dispersed corporation. The technique is based on a mesh of Real-time Transport Protocol (RTP) [4] relays. This is a hybrid communication technique that combines features of unicast reflectors with application-layer multicast. The RTP relays are used in a similar manner to User Datagram Protocol (UDP) reflectors, however, the RTP relays operate at the application level in the protocol stack, whereas UDP reflectors operate at the transport level. Since the relays understand RTP traffic, they can process RTP streams to encode data, mix multiple streams into a single stream, select which of several streams is to be forwarded, etc.

At present, the technique does not have an automatic method to deploy the communication mesh between the RTP relays. An example of the deployment of the communication mesh is shown in Fig. 1.

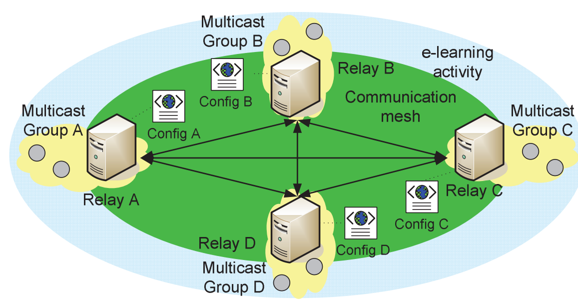


Figure 1: Communication mesh between the RTP relays.

The RTP relays are deployed using a full-mesh topology, where each RTP relay is connected to the other RTP relays. This mesh extends the scope of the e-learning activity between several dispersed sites, emulating the behavior of a single multicast group. The users of each site participate in the e-learning activity sending multimedia streams to their RTP relay using IP multicast. Subsequently, the streams are forwarded to the rest of the relays. The users of the other sites also receive the multimedia streams through their RTP

978-0-7695-4145-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ICSNC.2010.40

relays using IP multicast. One drawback of this architecture is that an RTP relay receives traffic even when there are no participants in its site. Static establishment of the communication mesh is performed locally in the RTP relay of each site. Each RTP relay has a local eXtensible Markup Language (XML) configuration file that specifies its peer relays in the mesh, and the IP address of the multicast group to which the relay is to be joined. Obviously, a priori knowledge of which RTP relays will be involved in the communications is necessary.

The aim of this paper is to design a technique for the automatic deployment of the communication mesh between the RTP relays. This technique optimally deploys full-mesh topologies during synchronous e-learning activities in the scope of corporate environments.

The remainder of this paper is organized as follows. In Section 2, related work on existing techniques for the deployment and management of communication meshes is discussed. The proposed technique is presented in Section 3. In Section 4, a strategy to validate the proposed technique is explained. Finally, Section 5 contains the concluding remarks and outlines future work.

II. Related Work

The techniques for the automatic deployment of a communication mesh among participating groups were surveyed by Tan et al. [5]. The authors analyzed the deployment and self-organization of ALMs, in two approaches: distributed transformation protocols, and localized central arrangement protocols. Using the distributed transformation protocols, nodes make independent decisions to reorganize the mesh; whereas with the localized arrangement protocols, a representative node is used to manage a centralized organization.

The most important technique in the scope of distributed techniques is Narada [6]. Narada builds a virtual mesh to link the participants. Fully distributed organization of the mesh is determined by the participants themselves, involving no external agent. The composition of the mesh is initially based on delay measurements between participants.

Another prominent work is NICE, developed by Banerjee et al. [7]. NICE assigns participants to different logical entities called layers. The participants in each layer are organized into clusters. The participant with the shortest distance from the rest is situated at the top of the cluster. The cluster and layers are built using a distributed algorithm. In a subsequent work [8], a primary node of the mesh has complete knowledge of the others. The primary node centrally creates the initial distribution tree. These techniques assume that native multicast communication is not available. However, IP multicast is usually available for synchronous e-learning in multinational corporations, so multiple RTP relays may be deployed as multicast proxies. Thus, it is only necessary to manage the mesh made up of a relatively small number of RTP relays. Since the RTP relays must have similar knowledge about the topology of the communication mesh, it is more appropriate to use a centralized technique. Distributed management of the mesh would require a high number of control messages. The most frequent method cited in the scientific literature for the centralized deployment and control of a communication mesh is the use of a Rendezvous Point (RP).

There are various ALMs that use an RP. Roca and El-Sayed propose a Host-Based Multicast technique (HBM) [9] for group communications. In HBM, end systems and dedicated servers automatically create a robust overlay topology to distribute data. HBM is based on an RP server, which manages the participants and the mesh topology. The HBM evaluates the distance between the RP and the rest of the participants using the transmission delay. It uses a shortest path tree (SPT) or a minimum spanning tree (MST) to build the initial topology of the communication mesh. It should be noted that the RP can combine two roles: the communication mesh controller and the source of the distribution tree.

Pendarakis et al. developed ALMI [10], an Application Level Multicast Infrastructure. ALMI provides a model involving a session handler and multiple participants. The session handler is easily accessed by the participants, running on a dedicated server. The handler estimates the minimum spanning tree in dialogue with all the participants, obtaining a series of round-trip delay measurements that characterize the performance of communications.

Zhang et al. [11] described a technique based on an HMRP (Host Multicast Rendezvous Point). The HMRP always knows which host is the root of the shared tree that represents the communication mesh. New participants query the HMRP about the root, and then try a small subset of participants as potential parents starting from root. The HMRP can be a single end host, a dedicated server, or a cluster of servers.

The aforementioned techniques are not appropriate for the requirements presented in this paper. The mesh communicating the RTP relays is always deployed in a full mesh topology, which has the lowest communication latency. These techniques depend on the behavior of the underlying network to provide minimal routing paths. Although they use the concept of RP, they are unsuitable as they cannot provide the required full-mesh topology.

Liu et al. [12] proposed an ALM technique called SDG (Sunflower Delivery Graph). This technique builds a shared tree among participants consisting of a series of key nodes, corresponding to data sources. Other participants connect to these key nodes. The initial structure of the tree is a full mesh topology, but it is progressively transformed into a tree using the minimum spanning tree algorithm.

Cicic et al. developed a communication technique based on UDP reflectors [13]. They present a system for web integrated control of the multicast-unicast reflectors. In this technique, the multimedia session descriptions and announcements are accessible using web browsers, and can be linked to the reflectors. They used the RTSP protocol to manage the sessions, but did not provide a self-deployment technique.

Hladka et al. [14] developed a UDP reflector associated with a web server. Through this server, it is possible to administer the reflectors manually, adding or removing reflectors. A full-mesh topology is used to connect the reflectors, while unicast connections are used between users and reflectors. Hladka et al. also developed a more flexible version of their initial UDP reflector [15]. This version implements a specific protocol, RAP, to communicate reflectors and users, although the deployment and reorganization of the reflector mesh is still manual.

None of the aforementioned reflectors have a self-deployment or self-organizing technique, so any change in the communication mesh must be done manually.

Patrikakis et al. implemented a streaming video distribution scheme based on reflectors, with its own automatic configuration and maintenance techniques [16]. They use an RP, the OCM server, to control the deployment and management of the mesh. The OCM server receives requests from participants and reflectors, and tells the participants which reflector they should connect to. The OCM server uses a permanent TCP connection to communicate with reflectors and detect failures as a keep-alive mechanism. This technique is tailored to activities of video streaming distribution. In this kind of activity, the source does not change during the activity. In contrast, in an e-learning activity the data source is subject to change.

Maraviglia et al. [17] proposed an architectural design to provide multipoint communications in a heterogeneous unicast/multicast environment. Their work is based on nodes, called S-MCUs, which are connected, constituting an overlay network of S-MCUs. Every S-MCU creates and maintains its own list of all the clients. They accept joining and leaving requests from clients and contribute to the creation of the overlay network. The communication mesh is centrally calculated by an external server based on information from each S-MCU. This external server also manages all the session details, including user accounts and permissions. This technique does not exclude the inactive sites, nor does it use a standard protocol to maintain the client list.

Reflex is a commercial RTP/SIP-based audio/video reflector and mixer for multi-point teleconferencing [18]. The Session Initiation Protocol (SIP) is used for the session setup signaling. This tool has a standard technique for the management of participants, but the configuration of the mesh of reflectors must be done manually.

This analysis indicates that there is no technique tailored to the requirements addressed in this paper.

III. PROPOSED TECHNIQUE

This section describes the proposed technique for the automatic deployment of a communication mesh between dispersed groups of participants in synchronous e-learning activities. Our proposal is tailored to the training of personnel in multi-national corporations and assumes the following:

- An RTP relay is located at each site of the corporation.
- Native IP multicast is available at each site.

We have developed a technique for automatically deploying the RTP relays associated with the sites involved in e-learning activities. This technique is based on several aspects of the techniques discussed above.

Automatic deployment of the mesh is supported by a central server, which is commonly called a Rendezvous Point (RP). This central server is accessible for all participants and RTP relays. The proposed technique consists of several stages that are take place chronologically, but which can be interlaced when RTP relays join or leave during an e-learning activity. The technique consists of a set of protocols and techniques discussed below, divided into the following stages:

- Registration of the RTP relays.
- Registration of the participants.
- Deployment and management of the communication mesh.

It should be noted that after completing these stages, two logical networks are established. The control network has a star topology, and is composed of all the RTP relays connected with the RP. An example of the control network is shown in Fig. 2. The data network has a full-mesh topology, as shown in Fig. 1. This network only covers the RTP relays. RTP relays forward multimedia streams over this data network.

A. Registration of the RTP Relays

Each RTP relay runs in a dedicated server, specifically as an operating system service. The RTP relay establishes a connection with the RP and sends a message to register itself in the RP. The RP has a web service which processes incoming registration requests, as shown in Fig. 2. The RTP relay provides the RP with an XML file, containing:

- The IP address and port for establishing the control channel between the RTP relay and the RP.
- The IP unicast addresses and port range for establishing the data channels between the RTP relay and its peer relays.
- The IP multicast addresses and port range for establishing the data channels between the RTP relay and its participants in ongoing activities.
- The identifier of the site where the RTP relay is located.
- The maximum number of participants supported by the RTP relay using multicast delivery.
- The maximum number of participants supported by the RTP relay using unicast delivery.

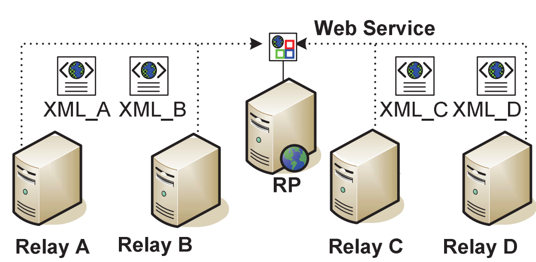


Figure 2: Registration of the RTP relays.

When the RP processes a registration message correctly, it responds to the RTP relay, indicating a successful registration. Hereafter, the RTP relay waits for messages from the RP reporting its peer relays in order to deploy the communication mesh between them. The RP maintains a list of the registered RTP relays and the sites in which they are located.

B. *Registration of the Participants*

The second stage involves the registration of participants in the e-learning activity. The RP will associate each participant with the RTP relay of the site where both are located. This stage involves two processes:

- Identification of the corporate site.
- Registration on the RP.

The participants must retrieve the identification of the site in which they are located before contacting the RP. As shown in Fig. 3, each RTP relay has an identification service bound to a multicast address that is known by all participants. Participants must send a multicast message to request information from the identification service.

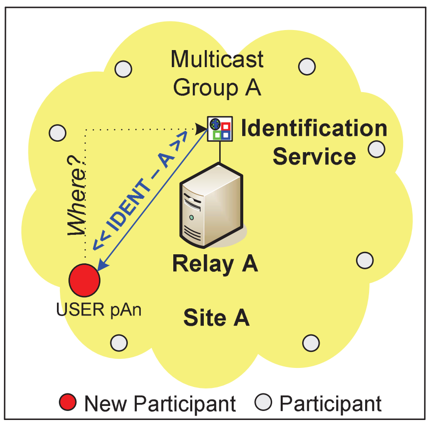


Figure 3. Requesting of site identification by the participants.

When an RTP relay receives an identification request, it responds to the participant with a message indicating the identifier of the site where the RTP relay is located. Note that the RTP relay does not store any information about the participants, as it merely responds to all identification requests received. Once the participant has identified the site from which it will take part in the e-learning activity, it must contact the RP. The RP plays the role of an SIP focus, as all participants must establish an SIP dialogue with the RP to enter an e-learning activity. The participant sends an SIP INVITE message to the RP in order to join an e-learning activity. The SIP message contains the identifier of the participant and the identifier of its geographical site. Fig. 4 illustrates the dialogue between the RP and the participant.

The RP associates all the registered RTP relays with the participants that have joined the e-learning activity from the site where the RTP relay is located. When the RP admits a new participant, it responds with a SIP OK message containing the description of the media that is transmitted during the e-learning activity. The description is based on the Session Description Protocol (SDP). IP multicast is used to transport data between the participant and its local RTP relay.

If there is no RTP relay available at the site from which a participant has requested participation in the activity, or if the identification process has failed, the participant uses a null identifier while contacting the RP. In this case, the RP associates the participant with an alternative RTP relay and the participant must use a unicast connection to send data traffic to the RTP relay. The unicast address of the relay is specified in the SDP description. Nevertheless, if the remaining RTP relays are assigned an excessively large number of participants, the RP will deny the entrance of a participant to the e-learning activity.

Once the participant processes the SDP description, it can join the activity. However, there are additional tasks that the RP must perform between the reception of the INVITE message from the participant and the sending of the SDP description, as discussed below.

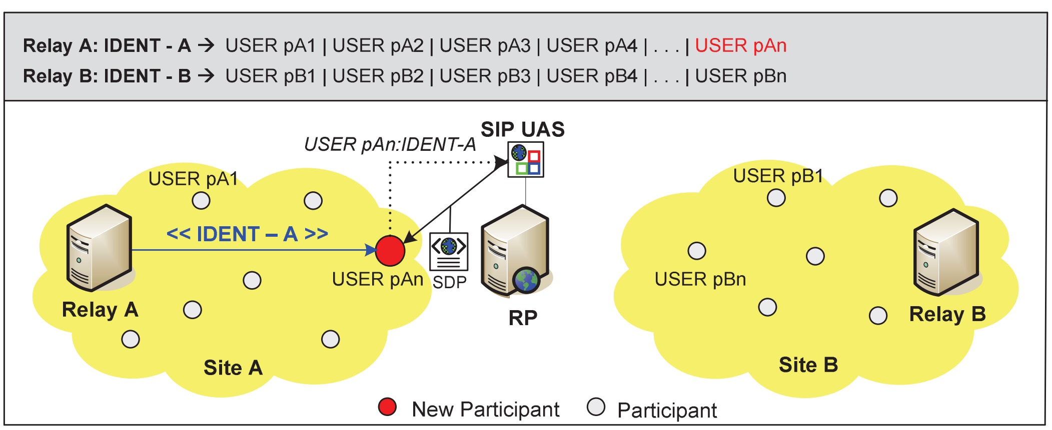


Figure 4. Registration of participants on the RP.

C. Deployment and Management of the Communication Mesh

As previously mentioned, the RP maintains a list of the registered RTP relays, associating them with the identification of the site in which they are located. A relay is active if at least one participant joins the e-learning activity from the site where the relay is located. Thus, the communication mesh only involves active relays.

Next, we describe the two basic processes in the management of the communication mesh: the inclusion and the exclusion of an RTP relay.

Fig. 5 illustrates the inclusion of an RTP relay in the communication mesh. The relay is included in the mesh when it becomes active. Fig. 5a shows a situation with four registered relays in the RP. There are two participants in the e-learning activity located at the site of relay A (relay A is active). A new participant from site B contacts the RP to join the ongoing activity. Therefore, as shown in Fig. 5b, relay B becomes active and all relays must be informed of the new mesh composition. The relay mesh is established between relays A and B. The RP sends an inclusion message to notify all relays that a new relay has been included in the communication mesh. When relay A receives the inclusion message, it must insert the peer relay (IP address and port) included in the message into its redistribution list. This is illustrated in Fig. 5b (left message). The IP received refers to a previously registered relay that has just become active.

The inclusion message sent to relay B contains as many peer relays as were already part of the mesh. This is illustrated in Fig. 5b (right message). The RP generates inclusion messages according to the information provided by relays during the registration process. Fig. 5c shows the communication mesh deployed between sites A and B after the processing of the inclusion messages.

Fig. 6 illustrates the messages used for excluding an RTP relay from the communication mesh. A relay is excluded from the communication mesh when it becomes inactive, that is, no participant joins the e-learning activity from the site where the relay is located. The rest of the relays must be instructed to stop forwarding data to the inactive relay. In Fig. 6a, the participant of site B leaves the e-learning activity. As a result, the RP sends an exclusion message to relay B, as shown in Fig. 6b. The relay processes the exclusion message and disconnects from the communication mesh. In addition, an elimination message is sent by the RP to all active relays (A) to notify that relay B has become inactive, as shown in Fig. 6b. The final state of the communication mesh is shown in Fig. 6c. The mesh is reduced to site A, so relay A does not need to forward data to other relays.

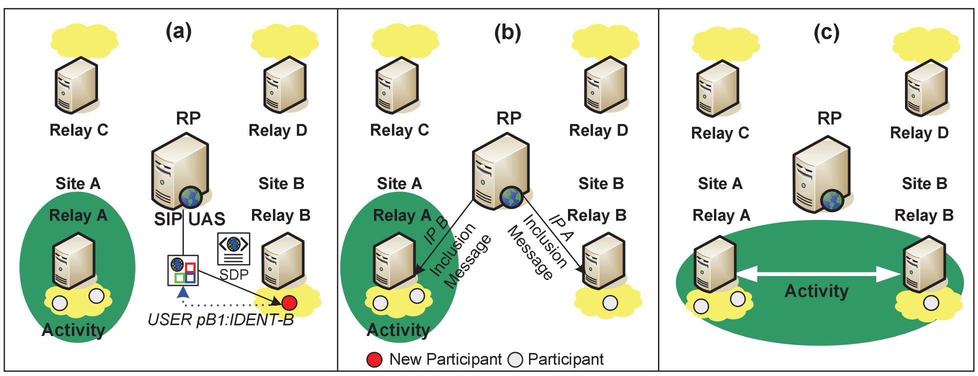


Figure 5. Management of the communication mesh – Inclusion of an RTP relay.

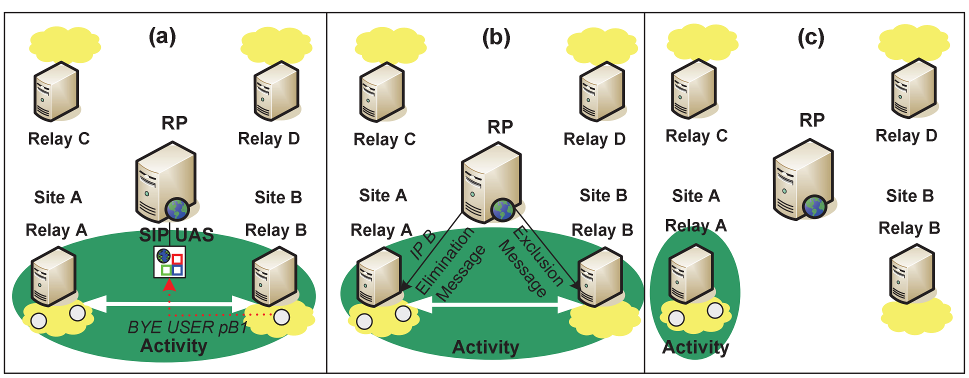


Figure 6. Management of the communication mesh – Exclusion of an RTP relay.

The communication mesh between the RTP relays is centrally managed by the RP. The RP is responsible for identifying the cases in which the joining or leaving of a participant in the e-learning activity implies a modification of the organization of the communication mesh. It is also responsible for notifying all the relays of the organizational changes, including and excluding them from the activity as explained above. This process ensures a minimum number of RTP streams forwarded between sites, since sites with no participants in the e-learning activity are excluded.

Fig. 7 shows the SIP messages exchanged between a participant and the RP during joining and leaving an e-learning activity. Initially, the participant sends an INVITE in order to join the activity. The interval labeled as Reconfiguration A is the period in which the topology of the mesh may change as a result of a new participant joining the activity. If this participant is the first in the site, the communication mesh will be modified to include the RTP relay of the site as shown in Fig. 5. Similarly, the interval labeled as Reconfiguration B is the period in which the topology of the mesh may be modified as a participant leaves the activity. If this participant is the last remaining in a site, the communication mesh will exclude the relay of this site as shown in Fig. 6.

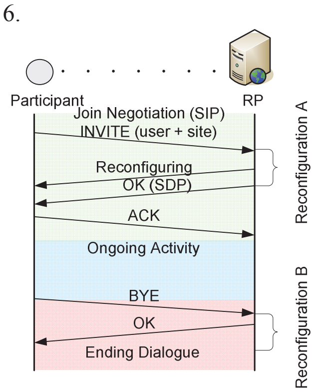


Figure 7. Messages exchanged between participant and the RP.

Fig. 8 details the states and transitions between states of an RTP relay, as well as the messages received from the RP. The first state is *Initialized*. In this state the relay is running, but is still not registered in the RP. Next, the relay sends the registration XML message to the RP and its state changes to *Awaiting Registration*. When the RP successfully processes the XML, the relay shifts to *Registered* state. If the registration process is not successful, the relay returns to the *Initialized* state. Once a relay is registered, it starts the identification service.

A participant joining an e-learning activity must use the identification service before contacting the RP. When the first participant of a site is accepted by the RP, the RP sends an inclusion message to all the active relays of the communication mesh to inform them of the inclusion of a new relay. Once the RP indicates its inclusion in the mesh, it becomes *Active*. In the *Active* state, the relay forwards traffic between participants in its site and its peer relays.

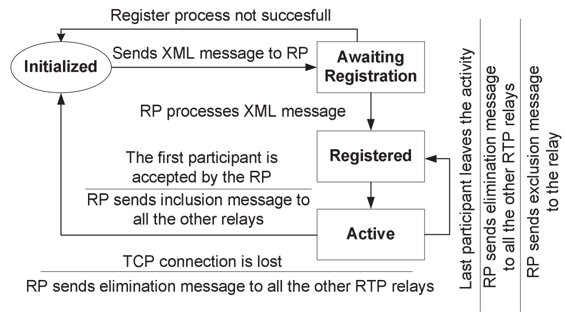


Figure 8. State diagram of an RTP relay during its operational period.

A relay in the *Active* state may move to the *Registered* state when it becomes inactive, that is, the last participant supported by the RTP relay leaves the e-learning activity. The relay receives an exclusion message from the RP indicating this situation.

The RP keeps an open TCP connection with each active relay providing support to the e-learning activity. This TCP connection is used as a keep-alive mechanism. The RP eliminates a relay from the communication mesh when its TCP connection is lost. In this case, the relay moves from the *Active* state to *Initialized*. The relay must repeat the registration process (stage A). This TCP connection is open from the start of the e-learning activity until the activity finishes, so the RP maintains updated information of active relays.

IV. Validation Strategy

The goal of the validation process is to assure that the automatic deployment of the communication mesh is correctly executed. The self-adaptation of the mesh to changes caused by the joining and leaving of participants is also validated. The validation tests have been carried out in an experimental environment that emulates the operation of the network of a geographically dispersed corporation.

A. Experimental Environment

In a real situation, the communication mesh is deployed in a corporate network as shown in Fig. 9. The corporation is dispersed in multiple sites, and there is an RTP relay in each site. Multicast communication is available in the LAN of each site. Participants usually join the e-learning activities from their workplaces, so they are within the multicast scope. All the active RTP relays are logically interconnected in a full-mesh topology (Fig. 1), so a path between two participants located in two different sites is always two relay hops in length. Finally, there is an RP managing the deployment of the relay mesh, as seen in Fig. 9. The RP receives requests from relays and participants, and is responsible for maintaining the composition of the communication mesh. However, due to the complexity of the real corporate environment, a simplified networking environment has been used to carry out the validation of the technique. The testbed environment (Fig. 10) emulates the corporate network explained above.

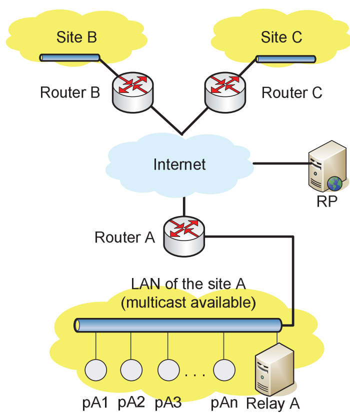


Figure 9. Real environment.

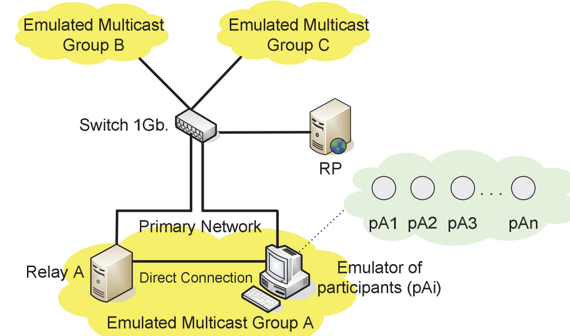


Figure 10. Simplified experimental environment.

A gigabit Ethernet switch is used to emulate the whole network (Internet) between all the sites. Each site is emulated with two computers that establish a multicast communication between them: one operates as the RTP relay and the other emulates all the participants of the site in the e-learning activity. Each site is assigned a different multicast group, so multicast traffic within each emulated site does not interfere with multicast traffic from another site. Multicast traffic between participants and their relay is conveyed through a direct link connecting the relay and the emulator of participants. In order to simplify the experimental procedure, the activity of each participant produces continuous transmission of an audio stream to all the other participants, as well as the reception of the streams sent by all the others.

Each RTP relay is equipped with two Network Interface Cards (NICs). The RTP relays are connected to the LAN network (primary network) using the first NIC. The second NIC is used to connect both the relay and the emulator of participants using multicast delivery. Each emulator of participants is also equipped with two NICs. The first NIC is used to send and receive SIP messages to and from the RP through the primary network. The second NIC is used to send and receive multicast data to and from the relay. An Ethernet crossover cable connects the secondary NICs of the relay and the emulator of participants. The relay uses its first NIC to forward data to its peer relays and to contact the RP.

Moreover, the experimental environment also includes an additional computer in the primary network playing the role of the RP. The RP only receives register messages from the RTP relays, and SIP messages from the emulated participants.

B. *Experimental Testing*

During the validation process, an emulated participant sends an audio stream to its RTP relay, encoded using iLBC. Audio is captured with a packet size of 20 milliseconds, resulting in a bitrate of 15.3 kbps. The worst scenario has been considered during tests, where each participant sends a single audio stream to all the others. The validation process checks two aspects of the proposed deployment technique.

Firstly, the communications mesh must be correctly deployed between any number of sites, and the number of control messages exchanged between the RTP relays and the RP must be low. Secondly, the mesh must be self-managed during the e-learning activity, with participants continuously joining and leaving the activity.

The deployment of the communication mesh is tested manually. The RTP relays are initialized sequentially with a short interval between them, and the registration process on the RP is verified. After registration, multiple participants join the activity from each site. Thus, it is possible to confirm that the RP is changing the topology of the communication mesh between RTP relays properly. Finally, when the communication mesh is totally deployed, the exclusion of the e-learning activity of all the participants of a site involves the re-organization of the configuration mesh by the RP. All the configurations of the communication mesh were registered in a trace during the tests, which was later checked to verify the accuracy of the technique.

The number of participants in the e-learning activity was changed from test to test, leaving active and inactive sites at random. This helped to confirm the successful deployment of the communication mesh making it possible to assess the traffic load in the communication mesh after each change. A theoretical model of the audio streams supported by an RTP relay at any time can be used.

C. *Theoretical Model of the TrafficSsupported by the RTP Relay*

In a corporation with N sites, the communication mesh can be composed of a maximum of N RTP relays. Nevertheless, only R sites have active participants. Let Si be the number of participants located in the same site as the relay i, which is responsible for forwarding RTP traffic coming from these participants to the peer relays and vice versa. Thus, Equation 1 represents the number of incoming audio streams at relay i coming from the rest of the relays and from the participants in the same site.

$$\sum_{j=1}^{R} S_j \qquad (1)$$

Equation 2 represents the number of outgoing audio streams at the RTP relay *i*.

$$S_i \times (R-1) + \sum_{j=1, j\neq i}^{R} S_j \qquad (2)$$

The left operand in (2) represents the audio streams that relay *i* sends to the other relays using unicast connections. The right operand in (2) represents the audio streams coming from the rest of the relays that the relay must forward to its multicast group.

V. Conclusions and Future Work

A centralized technique for the automatic deployment of a communication mesh between dispersed multicast-capable sites based on a Rendezvous Point (RP) has been proposed. The main advantage of using a centralized technique instead of a distributed technique is the minimal exchange of control information between participants. The use of an RP eliminates the need for a human operator to manually configure the communication mesh before starting the e-learning activity.

Furthermore, the proposed technique updates the composition of the mesh reacting to events generated during an e-learning activity, excluding those sites with no active participants, minimizing the traffic in the physical network.

This technique ensures self-deployment and self-management of the communication mesh, based on the joining and leaving of participants in the activity, using standard protocols such as SIP and XML.

The drawback of this technique is that the RP may fail or the connection between the RP and the relays may be interrupted. This would impede changes in the composition of the communication mesh, and would not allow the participants to join or leave the activity in a controlled manner, although the ongoing activities may continue. Therefore, this technique requires the RP to run in a fault tolerant cluster, and that the RP has redundant networking equipment.

Future work will be oriented to increasing the capability of the RP to support multiple e-learning activities concurrently. Another research project is how to recover from an RP failure without reconfiguring the communication mesh.

References

[1.] D. F. García, C. Uría, J. C. Granda, F. J. Suárez, and F. González (2007). "A functional evaluation of the commercial platforms and tools for synchronous distance e-learning," NAUN International Journal of Education and Information Technologies, vol. 1(2), pp. 95–104.

[2.] El-Sayed, V. Roca, and L. Mathy (2003). "A survey of proposals for an alternative group communication service," IEEE Network, vol. 17(1), pp. 46–51.

[3.] J. C. Granda, D. F. García, P. Nuño, and F. J. Suárez (2009). "Networking technique for synchronous e-learning platforms in corporate environments," in Proc. Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems. pp. 96–103, Istambul, Turkey.

[4.] Internet Engineering Task Force. "RTP: a transport protocol for real-time applications," RFC 3550, <http://www.ietf.org/rfc/rfc3550.txt> accessed in may 2010.

[5.] S. Tan, G. Waters, and J. Crawford (2006). "A performance comparison of self-organising application layer multicast overlay construction techniques," Journal of Computer Communications, vol. 29(12), pp. 2322–2347.

[6.] Y. H. Chu, S. G. Rao, S. Seshan, and H. Zhang (2000). "A case for end system multicast," in Proc. ACM Int. Conf. on Measurement and Modeling Computer Systems, pp. 1–12, Santa Clara, USA.

[7.] S. Banerjee, B. Bhattacharjee, and C. Kommareddy (2002). "Scalable application layer multicast," in Proc. of ACM Sigcomm, pp. 19-23, Pittsburgh, USA.

[8.] S. Banerjee, C. Kommareddy, K Kar, B. Bhattacharjee, and S. Khuller (2006). "OMNI: an efficient overlay multicast infrastructure for real-time applications," Journal of Computer Networks, vol. 50(6), pp. 826–841.

[9.] Roca and A. El-Sayed (2001). "Host-based multicast (HBM) solution for group communications," in Proc. 1st Int. Conf. on Networking, vol. 2093 of LNCS, pp. 610–619, Colmar, France.

[10.] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel (2001)."ALMI: An application level multicast infrastructure," in Proc. 3rd Usenix Symposium on Internet Technologies and Systems, pp. 49–60, San Francisco, USA.

[11.] Zhang, S. Jamin, and L. Zhang (2002). "Host multicast - a framework for delivering multicast to end users," in Proc. of IEEE Infocom, vol. 3, pp. 1366–1375, New York, USA.

[12.] Liu, G. He, and Z. Liu (2005). "Metrino – an application-level multicast system for real-time video communication," in Proc. 1st Int. Conf. on Wireless Communications, Networking and Mobile Computing, pp. 1265–1270, Wuham, China.

[13.] T. Cicic, H. Bryhni, and S. Sørlie (2000). "Unicast extensions to IP multicast," in Proc. of the Protocols for Multimedia Systems, pp. 60–69, Krakow, Poland.

[14.] E. Hladka, P. Holub, and J. Denemark (2002). "Teleconferencing support for small groups", in Proc. TERENA Networking Conference, pp. 1–7, Limerick, Ireland.

[15.] E. Hladka, P. Holub, and J. Denemark (2004). "User-empowered programmable network support for collaborative environment", in Proc. 3rd European Conf. on Universal Multiservice, vol. 3262 of LNCS, pp. 367–376, Porto, Portugal.

[16.] Z. Patrikakis, Y. Despotopoulos, A. M. Robotis, and A. L. Lambiris (2003). "PERIPHLEX: multicast delivery using core unicast distribution with peripheral multicast reflectors," in Proc. 12th Int. World Wide Web Conference, pp. 292, Budapest, Hungary.

[17.] G. Maraviglia et al. (2007). "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, pp. 1172–1179, Patras, Grecia.

[18.] Alkit Reflex. M. Johanson, Alkit Reflex RTP reflector/mixer, White paper of Alkit Communications, <http://www.alkit.se/> accessed in may 2010.

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

J Netw Syst Manage
DOI 10.1007/s10922-013-9290-4

Autonomic Platform for Synchronous e-Training in Dispersed Organizations

Juan C. Granda · Pelayo Nuño · Daniel F. García · Francisco J. Suárez

Received: 12 November 2012 / Revised: 23 September 2013 / Accepted: 27 September 2013

Abstract Synchronous e-training of human resources uses conferencing services to support training at the workplace. Multimedia data is interchanged between participants in real-time in a synchronous e-training activity providing an enriched learning experience. Large organizations are usually widely dispersed in several sites, so synchronous e-training activities are especially difficult. Most of the sites constitute multicast islands, as IP multicast is available, so the data transport can use IP multicast within the sites of the organization in order to save bandwidth. In this paper, an autonomic synchronous e-training platform based on standard protocols implementing self-management features such as self-deployment, self-organization and self-healing is proposed. The platform deploys an overlay network composed of Real-time Transport Protocol relays acting as multicast/unicast reflectors. The overlay is automatically reorganized according to the joining and leaving of participants and network failures. The use of standard protocols and a modular design enable the interoperability, extensibility and portability of the platform.

Keywords Self-organization · Self-healing · Overlay multicast · Interactive multimedia · Session Initiation Protocol

J. C. Granda (✉) · P. Nuño · D. F. García · F. J. Suárez
Department of Computer Science and Engineering, University of Oviedo, Gijón, Spain
e-mail: jcgranda@uniovi.es

P. Nuño
e-mail: nunopelayo@uniovi.es

D. F. García
e-mail: dfgarcia@uniovi.es

F. J. Suárez
e-mail: fjsuarez@uniovi.es

Published online: 06 October 2013

1 Introduction

Synchronous e-training has emerged as a complement to traditional training in large organizations in recent years. Synchronous e-training platforms make intensive use of multimedia conferences involving employees at their workplaces [1].

Features such as audio and video, shared whiteboards, telepointers and collaborative annotation of documents are common in synchronous e-training tools, so a synchronous e-training platform must cope with the multimedia data transport between participants. Thus, a synchronous e-training platform resembles a conferencing platform. However, corporate conferencing usually involves several dispersed groups communicating within conferencing rooms, while users are located at their workplaces in synchronous e-training, so productivity and interactivity between users are improved. Users can send and receive multimedia data to and from all the other users in synchronous e-training activities.

Although recent advances in network technologies have increased the bandwidth of network links, network resources are still considered valuable assets within large organizations. This is especially true in dispersed organizations, as corporate data traverses public networks through virtual private network links and is not limited to the corporate network scope. As a result, traffic from synchronous e-training activities must share the available network resources with other data which is usually crucial for the daily tasks of the organization.

The aim of a synchronous e-training platform is to provide a reliable and highly efficient data transport service with minimum network resource consumption. The latency introduced in data delivery must be as low as possible, so communications can be interactive.

Native IP multicast satisfies all the aforementioned requirements for multipoint group communication [2]. In fact, IP multicast is the best data delivery scenario, as there is no data replication at the network level and it conveys data from sources to receivers with the lowest latency [3]. Network routers must be appropriately configured to support IP multicast. A successful experience in the use of IP multicast supporting very large conferences has been reported in [4]. However, since IP multicast has several security issues and may consume a large amount of network resources [5, 6], the availability of IP multicast is limited, as many Internet service providers do not enable IP multicast in their networks. Therefore, IP multicast is rarely available throughout corporate networks.

The IP multicast service can be emulated using different techniques but at the cost of data transport efficiency. These techniques usually imply the deployment of overlay networks. In addition to an efficient data delivery service, a synchronous e-training platform must be able to automatically configure the data transport overlay network between users. The platform must also adapt itself as network or user failures occur and when users join and leave the activity, exhibiting an autonomic behavior [7].

Finally, platforms in which management issues are not transparent to users are difficult to maintain. If the setup or management of a synchronous e-training activity requires human intervention, highly specialized staff is necessary and the preparation of the activity is a complex and expensive process, as well as being prone to errors. Furthermore, platforms based on ad hoc solutions not using standard protocols and recommendations are incompatible with other software or hardware conferencing solutions.

In this paper an autonomic platform for synchronous e-training in dispersed organizations based on standard protocols and several multicast/unicast reflectors is presented. The e-training platform implements self-management features such as self-deployment, self-organization and self-healing, providing an efficient and self-managed transport overlay network. The overlay is automatically reorganized whenever a user leaves or joins the activity or a network failure occurs, managing the underlying network resources efficiently to provide real-time communications. Therefore, the platform requires minimum human interaction to configure the overlay and set up activities. Furthermore, the use of standard protocols allows for interoperability with other systems and devices, so the extensibility of the platform is guaranteed.

The remainder of this paper is organized as follows. In Sect. 2, related work on existing techniques for real-time multimedia communication is discussed. The elements that constitute the synchronous e-training platform are described in Sect. 3. In Sect. 4, the design and the operation of the Rendezvous Point are exposed. The self-organization technique used to deploy the transport overlay network is explained in Sect. 5. The synchronous e-training platform is tested in Sect. 6 and the results exposed in Sect. 7. Finally, Sect. 8 contains the concluding remarks.

2 Related Work

Exhaustive analyses of the techniques used to interconnect distributed groups of users by emulating IP multicast services were carried out in [8, 9]. Multicast/unicast reflectors and Application Layer Multicast (ALM) are the most common. Both alternatives are appropriate for synchronous e-training activities, but their efficiency depends on the number of groups to interconnect, the maximum latency allowed or the number of data sources.

2.1 Multicast/Unicast Reflectors

A multicast/unicast reflector is a network entity that forwards incoming multimedia data to one or several destinations, which may be receivers or other reflectors. Thus, it is possible to deploy an overlay network based on reflectors.

One of the initial overlays based on reflectors is described in [10]. The authors propose an architecture enabling unicast users to join multicast groups and participate in communication sessions. An external session controller manages each reflector, determining if incoming users are directly accessible through a multicast network by exchanging Real-time Streaming Protocol (RTSP) messages. Each session controller manages a reflector using the reflector session protocol. This architecture is improved in [11] allowing the session controller to manage several reflectors, although self-healing techniques are not provided.

A technique using autonomous reflectors to maintain application-level connectivity in the case of network failures is developed in [6]. Reflectors decide to migrate, remove, clone or merge with other reflectors in a decentralized self-organized manner, in order to achieve low-cost tree configurations for the distribution of group data. However, each reflector can forward only one media stream (audio, video, chat, etc.).

A web portal to manage conferences and reflectors was developed in [12]. Each reflector includes a mechanism for detecting inactive participants based on timers. An extended version of the reflector is presented in [13], including authentication and authorization features. Self-organizing and self-healing techniques for managing the overlay are not considered.

In the streaming video distribution platform developed in [14] an external server, called Overlay Configuration Management (OCM) server, is used to control the deployment and management of the reflectors. The OCM server receives requests from participants and reflectors, and instructs the participants as to which reflector they should connect to. The decision is made according to criteria such as the round-trip time and the number of participants connected to the reflector. The OCM server uses a TCP connection to communicate with reflectors as a keep-alive mechanism, and is able to relocate participants after a reflector failure. This architecture is built over the RTSP protocol.

The RTSP protocol is combined with the Session Announcement protocol (SAP) to build a live streaming architecture in [15]. Two special entities manage this architecture. First, a SAP/RTSP server announces the availability of new multimedia content and handles client joining requests. Second, an external server, called Resource Manager (RM), builds and manages a three-level overlay topology composed of live media servers, clients and reflectors acting as intermediaries between media servers and clients. The RM decides which relay must serve each client and also reorganizes clients on the fly to improve client quality of service. The management of media servers also relies on the RM.

An architecture to provide multipoint communication in a heterogeneous multicast/unicast environment is proposed in [16]. This is based on open software routers working as H.323 multipoint control units, called S-MCUs, constituting an overlay of S-MCUs. Every S-MCU creates and maintains its own list of participants. They accept joining and leaving requests from participants and contribute to the creation of the overlay. The organization of the overlay is determined by an external server using the information from each S-MCU. This server also manages the session details, including user accounts and permissions.

Finally, a hybrid approach for multi-party videoconferencing supporting Peer-to-Peer (P2P) and reflector forwarding is presented in [17]. Both data forwarding approaches can be used. When using reflector forwarding, some overlay members with multicast capability are able to play the role of a reflector. They forward data from the multicast domain to unicast users and vice versa. They also forward traffic between unicast users. The session management is controlled by an external entity called videoconferencing service center. This manages the session initialization, joining and leaving of overlay members, and connection building between reflectors and unicast users.

2.2 Application Layer Multicast

An ALM is a networking architecture in which the delivery of data between endpoints and management tasks usually rely on endpoints themselves. This allows the emulation of different network services such as IP multicast regardless of the configuration and capabilities of the underlying network. ALM overlays have more flexibility and scalability than overlays based on reflectors, but the latency is increased. Some of them are based on P2P models. The main difference between P2P-based and non P2P-based ALMs is the knowledge about the overlay. Each member propagates its knowledge about the overlay to its peers in P2P-based ALMs. In contrast, a precise knowledge about all the overlay is necessary for proper data distribution in non P2P-based ALMs. The responsibility for maintaining this knowledge may lie in one or more endpoints.

An analysis of self-organizing ALM techniques is developed in [18]. The authors compare distributed and centralized protocols for overlay management. The nodes of the overlay make independent decisions to reorganize the overlay when using a distributed protocol. In contrast, centralized protocols require a representative node responsible for the control of the overlay. This element is usually referred to as the Rendezvous Point (RP) since it also serves as entry point to the overlay. An example of centralized ALM is Application Level Multicast Infrastructure (ALMI) [19]. The RP manages two networks simultaneously. It maintains a control network with a star topology with non-persistent connections, and builds the data delivery network between the participants by estimating the minimum spanning tree. A similar approach is used in [20] for real-time video communications. However, most of the ALMs follow a decentralized approach.

Decentralized ALM overlays, usually based on P2P, achieve higher scalability than centralized overlays. They can be deployed according to several topologies [21], trees [22] and meshes [23] being the most common. These topologies can be combined leading to hybrid topologies, where endpoints can be organized in various local meshes across a global distribution tree in clustered topologies [24], or several distribution trees can be overlapped building a multi-tree topology [25].

P2P-based ALMs can be further classified as structured and unstructured according to whether they are tightly or loosely controlled respectively [26]. In unstructured overlays, a member randomly chooses some members of the overlay as its neighbors, while in structured overlays the neighborhood relationships are established according to specific rules [27]. Some structured P2P-based ALMs allow for deploying overlays where members are organized through hierarchical relations [28, 29]. Unstructured P2P-based ALMs allow for deploying overlays with highly volatile neighborhood relations according to data availability or endpoints contribution [30, 31]. These overlays are mainly aimed at large scale live streaming and video on demand [32, 33]. Finally, some recent ALM overlays follow a mixed approach combining structured and unstructured features [34, 35].

2.3 Analysis and Discussion

Endpoints in an ALM continuously evaluate the performance of their connections to determine the optimal routing paths. They can modify the relations with their neighbors to obtain the most efficient distribution approach, so the scalability of the overlay is usually high. Nevertheless, in spite of this, the latency introduced in communications may be high, as data traverses several hops from sources to receivers (depending on the topology of the ALM), especially in P2P-based ALMs. As a result, ALMs are not convenient for developing real-time activities in large organizations with low latency. On the other hand, a multi-reflector technique is appropriate when the latency of communications has to be low. Minimum latency is achieved when using an overlay network with the reflectors deployed in a full-mesh topology, as data traverses two reflectors maximum.

Many of the solutions based on reflectors are appropriate under the conditions imposed by multimedia real-time activities. However, most of them lack mechanisms for the automatic deployment and initial organization of the overlay, so they must be manually configured by an advanced user before an activity begins. Similarly, when users join and leave an activity, or when a reflector goes down, the overlay is not reorganized by excluding and including reflectors automatically, so the delivery service may become inefficient. Moreover, since the number of distributed groups of users can be assumed to be relatively small, it seems appropriate to use techniques where the management of the reflectors is centralized. The advantage of a centralized management approach is its simplicity and a lower bandwidth consumption than a distributed approach, as the latter usually needs a larger number of control messages between overlay members. Therefore, a technique based on an RP seems appropriate.

Finally, another desirable characteristic is the use of standard protocols and recommendations. Signaling protocols such as the Session Initiation Protocol (SIP), RTSP and H.323, allow for a standard overlay management, so the extensibility and interoperability of the platform is guaranteed. However, RTSP is aimed at large scale live streaming and video on demand. In the case of synchronous e-training activities, protocols such as SIP or H.323 are more suitable, as they are designed for audio and video conferencing. Furthermore, a simple architecture is needed to promote flexibility and portability, so the use of the SIP protocol to manage the overlay is the best choice. The fact that SIP has been adopted as the signaling protocol for third generation mobile telephony also reinforces the suitability of this protocol. Mobile devices can easily be supported by an e-training platform when using SIP.

None of the aforementioned techniques for real-time communications satisfy all these requirements, so a new solution is proposed to be applied in synchronous e-training activities.

3 Architectural Design

The overlay of the proposed synchronous e-training platform is composed of three virtual networks as depicted in Fig. 1. First, a Real-time Transport Protocol (RTP)

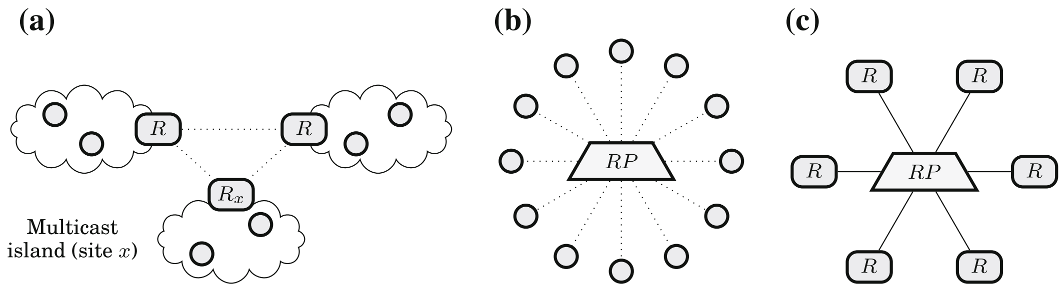


Fig. 1 Virtual networks composing the overlay: **a** relay mesh; **b** signaling network; **c** mesh control network

relay mesh is responsible for delivering multimedia data to participants in synchronous e-training activities. Each RTP relay acts as a multicast/unicast reflector of RTP traffic. Second, a signaling network allows users to participate in activities and facilitates the negotiation of the multimedia configurations of activities. Finally, a mesh control network is used to reorganize the relay mesh according to the joining and leaving of participants and network failures, so the real-time data delivery is efficient. The main advantage of this architectural design is modularity, which allows for developing a flexible and highly portable e-training platform. Thus, the topology and architecture of each virtual network can be changed without affecting the others.

3.1 RTP Relay Mesh

RTP is considered the de facto standard for delivering continuous media such as audio and video through IP networks [36], but it can also be used to transport other time-dependant data. Since multimedia data in a synchronous e-training activity must be delivered in real-time to enable interactive communications, RTP is ideal for synchronous platforms [37]. RTP runs over UDP/IP and separate RTP sessions must be used to deliver each media type. An RTP session represents an association of participants, which is not limited to a specific geographical location or a network scope, involving all the participants in an activity so they can send and receive data to and from the others.

Ideally, IP multicast is available in the underlying network and each RTP session is associated to an IP multicast group, so participants must join various multicast groups in order to receive all the multimedia data. However, the availability of IP multicast in corporate networks is limited. A corporate network can be considered as a group of IP multicast islands interconnected through a non-multicast-capable network. An RTP relay replicates incoming RTP data to participants and other relays. The use of several interconnected relays creates a relay mesh, in which each relay acts as a proxy between the IP multicast island where it is located and the rest of the relays in the mesh. Thus, the relay mesh transparently emulates the behavior of a multicast network to RTP applications despite the capabilities of the underlying network.

A relay combines unicast and multicast delivery, ensuring minimum bandwidth usage. Unicast is used to deliver data to peer relays, whereas IP multicast is used to distribute data to participants within the multicast island. Thus, the forwarding of the

RTP traffic is optimal and guarantees that the number of replicas of each RTP stream is minimal, providing an efficient delivery service [38]. Furthermore, since the relay understands RTP traffic, it is possible to process RTP to perform self-regulation techniques such as transcoding data, merging several RTP streams into one, selecting which stream to forward, etc. Occasionally, a relay may also forward traffic to participants without IP multicast connectivity by sending and receiving multimedia streams directly to and from the participants using unicast. This is useful when a participant joins an activity from outside the corporate network or the relay in the multicast island of the participant is down, so the participant can be redirected to another relay outside its multicast island.

All relays in the mesh are connected in a full-mesh topology, as shown in Fig. 1a, so as to minimize transport delay. Data from a participant must traverse a maximum of two relay hops. Each participant, except those from outside the corporate network, joins an activity from a multicast island where a relay is located, so the participant may be associated with the relay. As a result, participants can communicate with the others sending and receiving multicast streams. The use of a full-mesh topology makes the distribution tree trivial, since relays have to forward multicast data to all peer relays and unicast data coming from the rest of the relays to participants. Traffic must also be forwarded to participants outside their multicast island or connected from outside the corporate network and vice versa using unicast.

In addition to the transport protocol, RTP defines a complementary control protocol, the Real-time Control Transport Protocol (RTCP). All the participants in an RTP session transmit periodic RTCP packets within a variable interval of time which is recalculated according to the joining and leaving of participants to maintain scalability. RTCP provides feedback on the quality of the data delivery and allows participants to maintain a list of all the participants in the RTP session. Therefore, RTCP must also be forwarded by the relay mesh.

When an RTP or RTCP packet arrives, the relay must determine if its destination IP address is a multicast group or the unicast IP address of the relay. If the packet comes from a multicast group, it must be forwarded to all peer relays. Also, the packet must be forwarded to participants outside the multicast island of the relay. If the incoming packet is sent directly to the relay, it implies that the packet comes from a peer relay or a participant outside the multicast island. If the source IP address of the packet does not belong to a peer relay, the packet comes from a participant outside the multicast island and is replicated to all peer relays. In contrast, packets coming from a peer relay are not replicated to the rest of peer relays, but are forwarded to all participants outside the multicast island. A unicast packet, whether it comes from a peer relay or a participant outside the multicast island, is also forwarded to the participants within the multicast island.

3.2 Signaling Network

SIP is a signaling protocol for initiating, managing and terminating multimedia communications in IP networks [39]. Users exchange text-encoded messages to agree on a multimedia configuration. Various extensions have been proposed to the original SIP standard including call control services, presence control services, instant messaging, mobility and interoperability with other existing telephony systems. The negotiation of the multimedia configuration between SIP entities is supported by the Session Description Protocol (SDP) [40]. Although SIP is mainly applied to Voice over IP (VoIP), which usually involves only two users in the multimedia conference, it can be used to control multipoint conferences. A SIP entity, referred to as the SIP focus, maintains a SIP communication with each participant to implement a conferencing framework for tightly coupled conferences [41].

In the proposed platform, the SIP protocol is used for the management of synchronous e-training activities. An RP plays the role of a SIP focus to implement a tightly coupled conferencing service. Every participant establishes a SIP dialog with the RP while the activity is running. Thus, signaling communications between participants and the RP constitute a virtual network as represented in Fig. 1b.

Participants join an activity using the SIP standard handshake process, which makes the synchronous e-training platform inter-operable with most SIP applications and devices, such as VoIP phones. Standard authentication mechanisms can also be used with SIP. Moreover, the RP implements a SIP registrar server, so participants may register to upload their current locations. Therefore, it is possible to implement dial-out scenarios for joining an activity, where the RP invites a registered user to participate in the activity.

3.3 Mesh Control Network

The mesh control network is a virtual network established between the RP and the RTP relays during synchronous e-training activities. The RP uses TCP connections with the relays to notify changes in the topology of the relay mesh. These changes are mainly due to inclusions and exclusions of relays from the mesh as participants join and leave or when network failures occur. Figure 1c shows the connections established between the RP and the relays.

Relays must contact the RP as soon as possible to register with it, so the RP has an updated list of all the relays. It is not necessary that all the relays participate in an ongoing activity. In fact, a relay will not be part of the relay mesh if there is no participant in its multicast island. When an activity begins, the RP disseminates the organization of the relay mesh to all the relays involved in the activity. As a consequence, a relay may be in three states: unregistered, when it is not registered with the RP; registered, when it has registered with the RP; and active, when it is part of the relay mesh of an ongoing activity.

3.4 Theoretical Model of Traffic

Participants in a synchronous e-training activity interchange multiple types of data such as audio, video, annotations on a shared whiteboard, etc. Each type of data is delivered using its own RTP session. It is possible to predict the number of RTP streams that traverse each relay in the platform as a function of the number of relays and the number of participants in the mesh. An analytic model of the number of streams that are received and forwarded by each relay in the mesh for an RTP session is proposed in [38]. If the streams have a constant bitrate or an upper bound for their bitrate can be calculated, as occurs with audio and video streams, the equivalent network bandwidth consumption can be computed.

The model assumes some simplifications. First, participants are supposed to send a constant number of streams to the RTP session, which is quite usual for audio and video. Second, the relay uses IP multicast to communicate with multicast-capable participants, while it uses unicast to communicate with peer relays and participants outside the IP multicast scope. Third, signaling and control traffic are excluded from the model. Finally, the model predicts the traffic for a single RTP session. If several types of data are used in an e-training activity, the model should be applied to each session individually.

A relay is deployed at each site of an organization (R relays). Let S_i be the number of participants that send data to relay i, which is responsible for forwarding the streams to the rest of the RTP session. Let U_i be the number of participants communicating with the relay using unicast. These are further divided into participants that send and receive data (US_i) and participants that only receive data (UR_i):

$$U_i = UR_i + US_i$$

Since a relay receives all the streams of the RTP session, the number of incoming streams to the relay is defined as:

$$\sum_{j=1}^{R} S_j \tag{1}$$

The streams forwarded by relay i can be divided into three groups. First, the relay forwards incoming traffic to peer relays. These streams are delivered using unicast transport:

$$S_i \times (R - 1) \tag{2}$$

Next, the number of streams that the relay forwards to the unicast participants is defined as:

$$\sum_{j=1}^{R} (S_j \times U_i) - US_i \tag{3}$$

The third group is made up of the streams coming from peer relays and the unicast participants, which the relay forwards to the multicast group:

$$US_i + \sum_{j=1, j \neq i}^{R} S_j \tag{4}$$

Thus, the number of outgoing streams from the relay is the result of (2) + (3) + (4).

4 Rendezvous Point

The RP is the essential element for the deployment and management of the synchronous e-training platform. The RP manages the automatic organization of the communication mesh between relays modifying the mesh topology according to changes caused by the joining and leaving of participants. The RP is also responsible for recovering and reorganizing the relay mesh when a relay in the mesh goes down. If a relay goes down, the RP must redirect the participants in the multicast island of the relay to other relays, so they can continue sending and receiving multimedia data. To do so, the RP must collect information about relays and their associated participants in order to maintain the integrity of the relay mesh.

The RP is the entry point to the synchronous e-training platform. All relays must register with the RP before the beginning of a synchronous e-training activity. Registered relays may be included in the relay mesh of future or ongoing activities. Participants may optionally register to the RP using SIP REGISTER messages, making their locations public to the RP.

Registration requests from a relay contain information about the relay and its capabilities to support activities. The registration of the relay with the RP has an expiration time to provide some feedback concerning the availability of the relay. The relay must re-register with the RP when the registration expires. Once the RP starts an activity, the relay may be requested to participate in the relay mesh of the activity. Nevertheless, there may be registered relays not participating in the activity if there is no participant joining from the site where the relay is located.

Participants send a SIP INVITE to the RP to join an activity. The RP authenticates users and provides the multimedia configuration of the activity. This configuration depends on the site where the requesting participant is located. The participant is associated with the relay located in the same multicast island, so it uses IP multicast to communicate with the relay.

A participant outside a multicast island may also join an activity. The RP associates the participant with a relay according to the capabilities of the relays in the mesh. This requires additional resources at the relay to forward traffic to this participant, as the participant will communicate with the relay using unicast. In some cases, the RP may reject the joining request from a participant outside the corporate network or exclude participants from the e-training activity when their relays go down.

However, the centralized architectural design for the control of the overlay has some drawbacks. All the overlay relies on the RP, becoming a critical point of failure. The consequence of a failure is only one of the criteria that can be used to analyze the different service failure modes originated by a disruption [42]. Consequences affecting the availability, safety, confidentiality or integrity of the overlay may be classified as fully-disruptive (FD), semi-disruptive (SD) and non-disruptive (ND) [43]. Consequences can be further divided into operational and functional consequences. Operational consequences are those involving the overlay network as a whole entity, preventing participants from developing e-training activities. On the other hand, functional consequences are those affecting only the quality of experience of participants in the e-training activities.

A failure or a network disruption affecting the RP implies different functional and operational consequences according to the activity state. During an ongoing activity, an RP failure does not imply fully-disruptive functional consequences. The RP is not responsible for delivering data, so the failure does not prevent participants from interchanging data. Nevertheless, the self-organization and self-healing techniques cannot be triggered when a participant leaves the activity or when a relay fails. This may lead to a waste of resources, as data streams may be unnecessarily forwarded to relays with no participants. Therefore, RP failures in an ongoing activity lead to a semi-disruptive operational state.

An RP failure also implies that the availability of the overlay decreases. New participants cannot establish a SIP dialog with the RP, so they cannot join the e-training activity. Thus, an RP failure in a partially established activity prevents participants from taking part in it, leading to a fully-disruptive operational/functional state.

To overcome this scenario, fault-tolerant hardware solutions like mirroring and hot spare configurations can be applied to RP. Since the RP is unique in the platform, a solution based on mirroring or redundant hardware is feasible. This is commonly used in 24×7 bootstrap servers of many P2P overlay networks.

E-training activities can be carried out with minimum costs. The costs of the platform can be measured using several metrics: economical, resource consumption and management. Firstly, the economic cost of the platform comes from the equipment required for the RP and the relays. A relay can be run in a regular PC, while the RP requires fault-tolerant hardware to guarantee the resilience of the platform. Only one relay and the RP are required for small scale activities when using a star topology for the relay mesh. Secondly, the resources consumed by the platform depend greatly on the dispersion of participants as analyzed in Sect. 7. Finally, the self-management characteristics of the platform help to reduce the management costs and the requirements of specialized staff.

Moreover, security is critical in corporate synchronous e-training. Many issues may affect the confidentiality, integrity and availability of the platform as pointed out in [44]. The platform allows for the use of the Secure Real-time Protocol (SRTP) for the delivery of data to provide data confidentiality and integrity. However, various security issues remain open such as source authentication in multicast multimedia communications over SRTP with multiple sources. A participant may impersonate other participants. This issue can be alleviated using access control lists in the relays, so data is only forwarded from authenticated participants.

5 Self-deployment and Self-organization Techniques

This section describes the technique for the autonomic deployment and management of the overlay. Our proposal is tailored to geographically dispersed organizations and assumes that an RTP relay is located at each site of the organization and native IP multicast is available at each site. The technique consists of several stages that take place chronologically, but which can be interlaced as participants join and leave an activity or when relay or network failures occur. The technique can be divided into the following stages:

- Registration of relays.
- Joining and leaving of participants.

- Deployment and management of the relay mesh.

5.1 Registration of Relays

Once an RTP relay has started running, it establishes a temporary TCP connection with the registration service to register with the RP. The relay provides the RP with an XML document, containing the following information:

- The IP address and port for establishing the control channel between the relay and the RP.
- The IP unicast addresses and port range available for establishing the data channels between the relay and its peer relays in the mesh or participants outside its multicast island.
- The IP multicast addresses and port range for establishing the data channels between the relay and the participants within its multicast island.
- The identifier of the site where the relay is located.
- The maximum number of participants supported by the relay using IP multicast.
- The maximum number of participants supported by the relay using unicast.

When the RP processes a registration request correctly, it responds to the relay, indicating a successful registration. Then, the TCP connection is closed. The registration has an expiration time, so the relay must re-register with the RP periodically. Hereafter, the relay waits for the RP to reestablish the TCP connection. This means that the RP is requesting the relay to participate in an ongoing activity. In this case, the relay becomes active and receives an XML document from the RP with information about ongoing activities and its peer relays in the relay mesh. The TCP connection remains open while the activity is running.

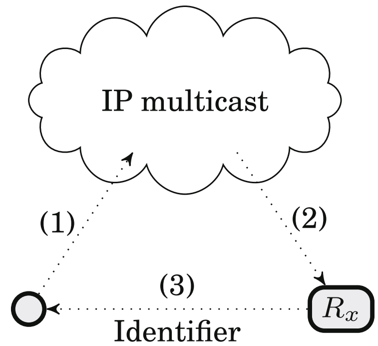
5.2 Joining and Leaving of Participants

A participant must contact the SIP focus module of the RP in order to join a synchronous e-training activity. If the RP accepts the joining request, it associates the participant with a relay in the relay mesh. This relay may have been part of the mesh previously, or the RP may have included the relay in the relay mesh as a result of the participant joining. In either case, the participant must obtain a site identifier, so the RP can associate the participant with the relay.

Two steps are identified when a participant joins an activity. Before joining an activity, the participant must be informed of the multicast island in which it is located by the corresponding relay. Then, the participant can join the activity by establishing a SIP dialog with the RP, indicating the identifier of the multicast island.

As shown in Fig. 2, each relay has an identification service bound to a multicast address that is known by all participants. Participants must send a multicast message to request information from the identification service. When a relay receives an identification request, it responds to the participant with a message indicating the identifier of the site where the relay is located. Note that the relay does not store any

Fig. 2 Identification of the multicast island



information about participants, as it merely responds to all identification requests received.

Once a participant has learned the site from which it will take part in the activity, it must contact the RP. The RP plays the role of a SIP focus, as all participants must establish a SIP dialog with the RP to join an activity. The participant sends a SIP INVITE message to the RP in order to join the ongoing activity. The SIP message contains the SIP Uniform Resource Identifier (URI) of the activity, the SIP URI of the participant and the identifier of its multicast island. The identifier of the multicast island is carried in an optional *Organization* header within the INVITE message.

The RP associates all the active relays with the participants that have joined the activity from the multicast island where the relay is located. When the RP accepts a new participant, it answers with a SIP OK message containing the description of the media that is transmitted during the activity. IP multicast is used to transport data between the participant and its local relay.

If there is no relay available at the multicast island from which a participant has requested participation in the activity, or if the identification process has failed, the participant does not include an *Organization* header in the INVITE message. In this case, the RP associates the participant with the relay with the least number of participants served and the participant uses unicast to send and receive data to and from the relay. The unicast address of the relay is specified in the SDP description. Nevertheless, if the remaining relays are assigned an excessively large number of participants, the RP will deny the entrance of a participant to the activity.

A participant leaves the activity by sending a SIP BYE message. The RP may also exclude a participant from the activity with a SIP BYE message. In the case of a participant leaving the activity ungracefully, the relay associated with the participant detects that the participant has left the activity, as the relay snoops RTCP traffic and a timer is used to timeout participants. Thus, the relay can notify the RP of participants leaving the activity.

5.3 Deployment and Management of the Relay Mesh

As previously mentioned, the RP maintains a list of the registered relays, associating them with the identifier of the multicast island in which they are located. A relay is active if at least one participant joins a synchronous e-training activity from the multicast island where the relay is located. Thus, the relay mesh only involves active relays. Next, the two processes involved in the management of the relay mesh are described: the inclusion and the exclusion of relays.

Figure 3 illustrates the inclusion of a relay in the relay mesh. The relay is included when it becomes active. Figure 3a shows a situation with four registered relays with the RP. A new participant from multicast island a establishes a SIP dialog with the RP to join the ongoing activity. Therefore, as shown in Fig. 3b, relay R_a becomes active and all relays must be informed of the new mesh organization. The relay mesh is established among relays R_a, R_c and R_d. The RP sends an inclusion message to notify relays R_c and R_d that a new relay has been included in the relay mesh. When relays R_c and R_d receive the inclusion message, they insert the peer relay included in the message into their redistribution lists. The RP also sends an inclusion message to relay R_a with information about all the active relays in the mesh. The inclusion messages contain information such as the IP address and a range of ports of one or several peer relays. The RP generates inclusion messages according to the information provided by relays during the registration process. Figure 3c shows the relay mesh deployed among multicast islands a, c and d after processing the inclusion messages.

Figure 4 illustrates the process carried out to exclude a relay from the relay mesh. A relay is excluded when it becomes inactive. The rest of the relays must be instructed to stop forwarding data to the inactive relay. In Fig. 4a, the participant from multicast island d leaves the activity. As a result, the RP sends an exclusion message to relay R_d, as shown in Fig. 4b. The relay processes the exclusion message and disconnects from the relay mesh. In addition, a removal message is sent by the RP to all active relays (R_a and R_c) to notify that relay R_d has become inactive. The final organization of the relay mesh is shown in Fig. 4c.

The RP is responsible for identifying the cases in which the joining or leaving of a participant in the activity implies a modification of the organization of the relay mesh. This ensures a minimum number of RTP streams forwarded between multicast islands, since those with no participants in activities are excluded.

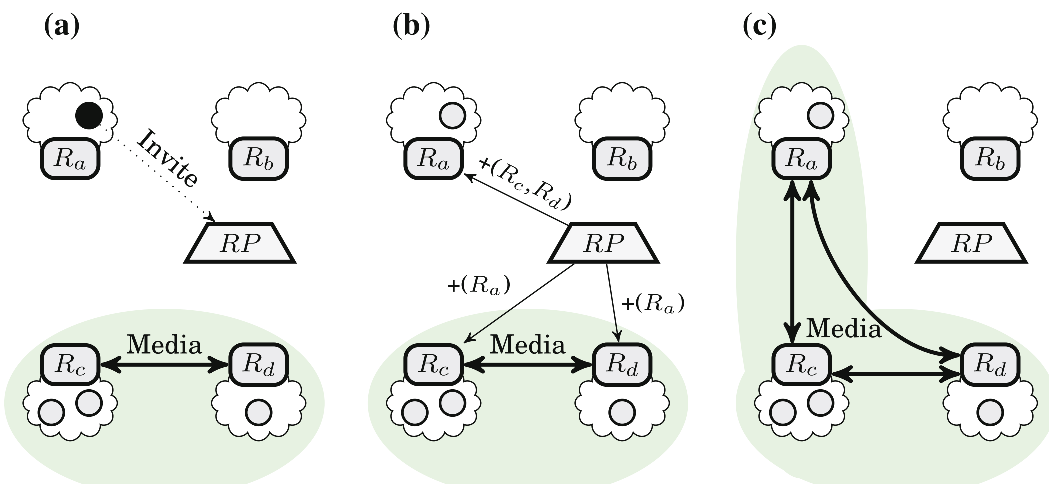


Fig. 3 Inclusion of a relay in the relay mesh of a synchronous e-training activity

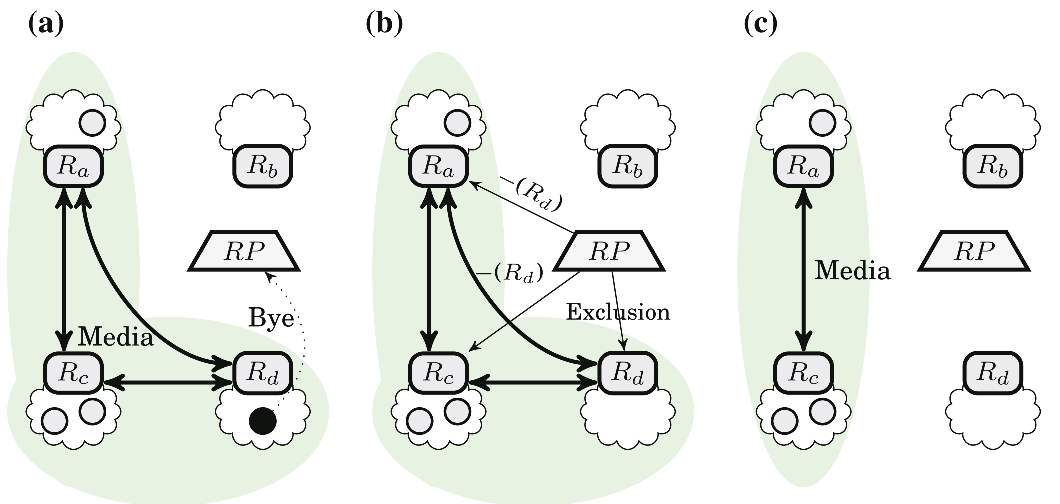


Fig. 4 Exclusion of a relay from the relay mesh of a synchronous e-training activity

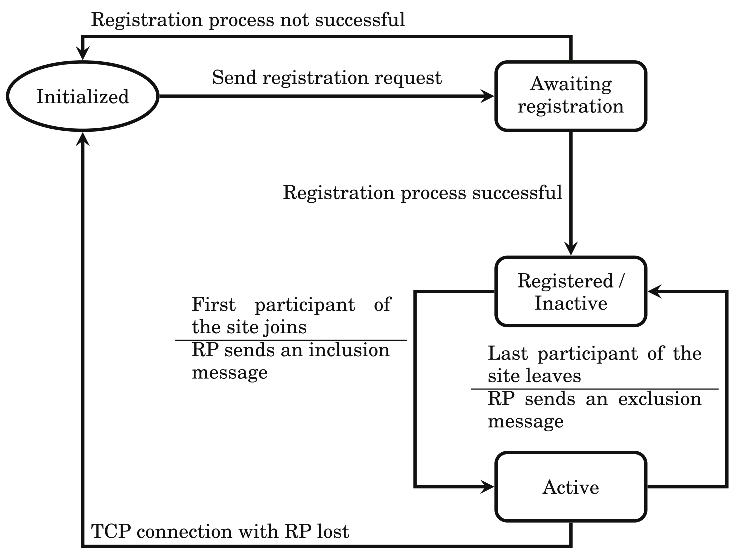


Fig. 5 State diagram of an RTP relay

Figure 5 details the states and transitions between states of a relay as a consequence of the messages received from the RP. The first state is *Initialized*. In this state the relay is running, but it is still not registered with the RP. Next, the relay sends the registration XML message to the RP and its state changes to *Awaiting Registration*. When the RP successfully processes the XML and the relay receives the confirmation, the relay shifts to *Registered* state. If the registration process is not successful, the relay returns to the *Initialized* state. Once a relay is registered, it starts the identification service.

When the first participant of the multicast island of the relay is accepted by the RP, the RP sends an inclusion message to the relay with information about the current organization of the relay mesh. Once the inclusion message is processed, the relay becomes *Active*, forwarding traffic to peer relays and participants.

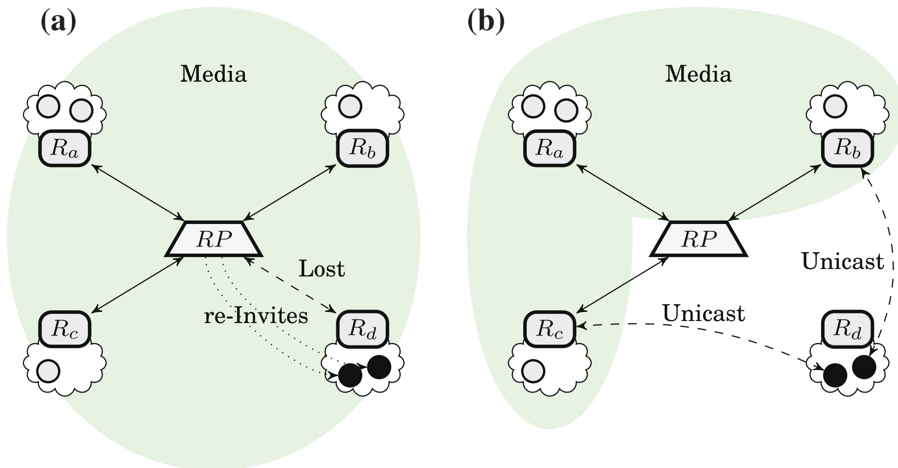


Fig. 6 Self-healing technique

A relay in *Active* state moves to *Registered* state when it becomes inactive, that is, the last participant supported by the relay leaves the ongoing activity. The relay receives an exclusion message from the RP indicating this situation.

The RP keeps an open TCP connection with each active relay. This TCP connection is used as a keep-alive mechanism. The RP eliminates a relay from the relay mesh when its TCP connection is lost. In this case, the relay moves from *Active* state to *Initialized* and it must repeat the registration process.

5.4 Self-healing Technique

A self-healing technique has been implemented in the platform that allows the continuity of the service in spite of relay failures. Whenever a relay goes down, the RP associates the participants of the relay to another active relay in the relay mesh. In this way, the participants still send and receive traffic to and from ongoing activities.

As presented above, the RP keeps an open TCP connection with each active relay and maintains updated information about the identity of participants in each relay. Therefore, when a TCP connection is lost, the RP knows the participants that must be redirected to another relay. The redirection process is illustrated in Fig. 6. Figure 6a shows an activity with four active relays. When the TCP connection between the RP and relay R_d is lost, the RP sends SIP re-INVITE messages to the two participants of relay R_d to change their configuration for data delivery. These SIP messages contain SDP descriptions which redirect traffic from the participants to other relays using unicast. Specifically, a participant is redirected to relay R_c and the other to relay R_b. The final organization of the relay mesh is shown in Fig. 6b.

When the relay is up again a self-optimization technique takes place. The relay registers with the RP and the redirection process is undone. The RP maintains a list with the changes made in the relay mesh, so it is able to roll back any of them.

The redirection process has some influence on the performance of the overlay. Although the number of incoming RTP streams remains the same for all the relays, the number of outgoing RTP streams increases in those relays to which participants are redirected. Let U_{ri} be the number of redirected participants to relay i. These participants can further be divided into participants sending and receiving data (US_{ri}) and participants that only receive data (UR_{ri}). The number of streams coming from the peer relays of relay i decreases by US_{ri} streams, but these streams are now received directly from unicast participants. On the other hand, the additional outgoing streams can be calculated as:

$$US_{ri} \times [(R-1) + (U_{ri}-1) + U_i + 1] \tag{5}$$

The streams generated by the redirected participants (US_{ri}) must be forwarded to the peer relays, the redirected participants, the rest of unicast participants and the multicast island of the relay. Taking into account the definition of redirected participants ($U_{ri} = US_{ri} + UR_{ri}$), Eq. (5) can be expressed as follows:

$$US_{ri}^2 + US_{ri} \times (UR_{ri} + U_i + R - 1) \tag{6}$$

Thus, the upstream bandwidth consumption at the relay increases exponentially with the number of redirected participants sending data.

6 Experimentation

The e-training platform has been completely implemented and is fully functional. It has been successfully used to support real synchronous e-training activities within a steel making company [37]. However, the assessment of the performance, robustness and resilience of the overlay requires exhaustive tests that cannot be carried out in a production environment. A model has been developed using the ns-3 simulator to perform the tests. The model simulates the operation of the network of a geographically dispersed organization and all the entities of the platform, including participants, relays and the RP.

6.1 Network Model

The network model simulates the network of an organization dispersed in multiple sites where IP multicast is available, and there is a relay in each site. An ideal Wide Area Network (WAN) connects every site with the RP through 20 Mbps network links with a delay of 20 ms. The data rate of the network link connecting the RP to the WAN is 10 Mbps and introduces no delay in communications.

The range of possible overlay deployments is extremely high. Many parameters such as the number of relays in the mesh, the number of participants, the balance of participants between relays and the number of unicast participants can be varied, resulting in different throughput results. Only one parameter is varied at a time in the tests in order to evaluate the effect of each parameter individually.

Moreover, it must be noted that the full-mesh deployment of the overlay might make the relay mesh incur in scalability issues for a high number of participants, since data replication grows with the number of relays and participants. Nevertheless, real-time interactive communication services usually involve small groups of participants, so simulations with a few dozen participants are representative of the environments in which the platform can be used. A complete assessment of the scalability of the relay mesh is carried out in [38].

6.2 Participant Modeling

All the participants are modeled in order to simulate the traffic generated during the synchronous e-training activities. A participant can use several kinds of media types to communicate with other participants in an activity. Typically, audio and video are the most commonly used media types, but other media types such as shared whiteboard annotations and telepointers can also be used. Thus, a participant is represented as an entity that generates audio and video streams, annotations in the shared whiteboard and telepointer movements. All this information must be forwarded to the rest of the participants.

A participant waits before joining the activity. The waiting time is simulated using an exponential random variable $Exp(\lambda_w)$. The participant establishes a SIP dialog with the RP in order to join the activity and negotiate the media configuration, and closes the dialog when leaving gracefully. Once the participant has joined the activity, he or she remains joined until the end or leaves unexpectedly without notifying the RP with a preset probability p_l.

The audio and video streams of a participant are activated regularly. A video stream is a simulated 160×120 variable bitrate H.264 video stream at 10 frames per second. An audio stream is a simulated constant bitrate iLBC audio stream with a packetization time of 20 ms. Both the interval between activations and the duration of the streams are assumed to be normal random variables. Thus, a participant generates an audio stream of duration $\mathcal{N}(\mu_{al}, \sigma_{al}^2)$ after an elapsed time $\mathcal{N}(\mu_a, \sigma_a^2)$, and generates a video stream of duration $\mathcal{N}(\mu_{vl}, \sigma_{vl}^2)$ after an elapsed time $\mathcal{N}(\mu_v, \sigma_v^2)$.

The format of the annotations in the shared whiteboard and the telepointers used by participants are described in [37]. A participant generates an RTP stream containing their annotations in the shared whiteboard, while another RTP stream is used to convey the information of the telepointer. Similarly to audio and video, the activation time and the duration of these streams are assumed to be normal random variables. Thus, a participant generates an annotation stream of duration $\mathcal{N}(\mu_{wl}, \sigma_{wl}^2)$ after an elapsed time $\mathcal{N}(\mu_w, \sigma_w^2)$, and generates a telepointer stream of duration $\mathcal{N}(\mu_{tl}, \sigma_{tl}^2)$ after an elapsed time $\mathcal{N}(\mu_t, \sigma_t^2)$.

Obviously, real activities are usually moderated in order to manage the interactions among participants and avoid excessive network resource consumption due to many multimedia streams. Floor control protocols can be used to share the data channels between the participants of an activity. The maximum number of data streams that can be activated simultaneously depends on the desirable interactivity for the activity. Since this number is difficult to establish as floor control policies depend greatly on the kind of activity to be carried out, three scenarios are considered in the tests: low, medium and high interaction. In a low interactive activity only 2 audio streams and 2 video streams are allowed simultaneously. In a medium interactive activity these numbers increase to 4 audio streams and 4 video streams. Finally, 4 audio streams and 6 video streams are allowed in a highly interactive activity. The maximum number of simultaneous annotation and telepointer streams are 2 regardless of the type of activity.

Two roles can be identified in the participants of a synchronous e-training activity: one instructor and many regular participants. The instructor is a participant who uses all the media types continuously to emulate the speech of the instructor during the activity. The rest of the participants issue floor requests to use the data channels. This situation closely resembles the behavior of users in synchronous e-training activities where participants usually interrupt the activity to ask questions and the instructor can be seen and heard throughout the duration of the activity. In all cases, the requests from participants to use the data channels are granted in a first-in first-out order.

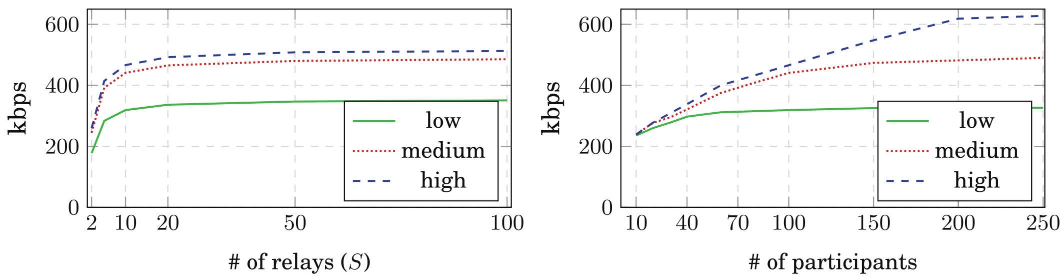
7 Results

Simulations were conducted to assess the performance and the self-healing ability of the overlay. Table 1 shows the settings used during the simulations. Each test simulated a balanced synchronous e-training activity of 1 hour where each relay serves the same number of participants and was repeated 7 times.

Figure 7 compares the average bandwidth consumption in the network links connecting the sites to the WAN depending on the floor control policy used. Figure 7a shows the bandwidth used in an activity with 100 participants when increasing their dispersion (the number of relays in the mesh), while the number of participants is varied for an activity with 10 relays in Fig. 7b. As can be seen, the traffic rapidly grows with the dispersion of participants, but the growth becomes asymptotic due to floor control policies. However, although the average bandwidth consumption is limited, some relays may consume significant bandwidth when senders are co-located in the same sites. To alleviate this issue, floor control decisions can be taken to balance senders between sites.

Table 1 Simulation settings

Waiting time to join (s)	λ_w	1/45
Unexpected leaving probability	p_l	0.05
Audio activation interval (min)	(μ_a, σ_a)	(25,3.3)
Audio stream duration (s)	(μ_{al}, σ_{al})	(15,2.6)
Video activation interval (min)	(μ_v, σ_v)	(40,8.2)
Video stream duration (s)	(μ_{vl}, σ_{vl})	(30,9.5)
Annotation activation interval (min)	(μ_w, σ_w)	(16.6,1.3)
Annotation stream duration (s)	(μ_{wl}, σ_{wl})	(60,10)
Telepointer activation interval (min)	(μ_t, σ_t)	(20,1.6)
Telepointer stream duration (s)	(μ_{tl}, σ_{tl})	(15,2.5)



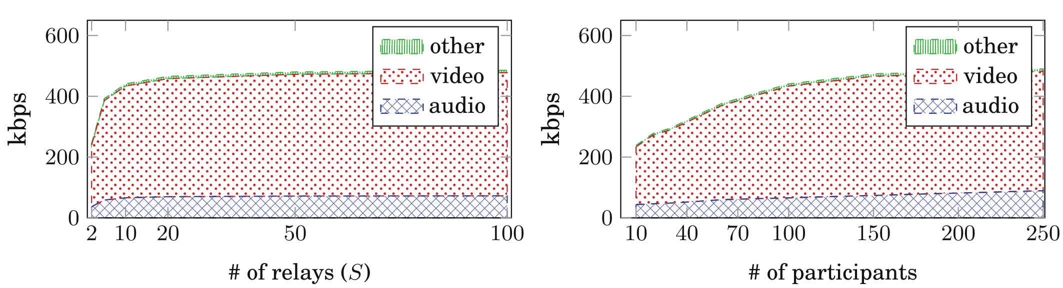
(a) 100 participants **(b)** 10 relays

Fig. 7 Average network bandwidth consumption in the network links of the sites depending on the interactivity of the activity

The traffic also grows asymptotically with the number of participants, although with a lower pace as depicted in Fig. 7b. For activities with a low number of participants the results are similar for all the interactivity scenarios, since not all the floors are granted simultaneously in higher interactive scenarios and a similar number of data streams are generated by participants. The small differences in the traffic observed between the three interactivity scenarios are due to control messages. The RTP session bandwidth used to compute RTCP transmission intervals depends on the maximum number of simultaneous floor holders, so a high interactivity level also implies additional RTCP traffic. However, when increasing the number of participants, the bandwidth used in the three interactivity scenarios diverges as more floors are held simultaneously in higher interactivity scenarios and more data streams flow throughout the platform.

The bandwidth consumption for the medium interactivity scenario is broken down in the stack area chart of Fig. 8. Figure 8a shows that video is clearly the most resource-consuming media type, while the average bandwidth consumed by annotations and telepointers is negligible compared to audio and video. On the other hand, two regions can be identified for each media type in Fig. 8b. The bandwidth consumption steadily grows with the number of participants initially when the average number of floors granted is less than the maximum. Once the maximum number of floors granted is reached, the bandwidth consumption increases asymptotically due to control messages from new participants.

The bandwidth consumption for the medium interactivity scenario is also compared to a delivery service based on unicast with the same floor control policy



(a) 100 participants **(b)** 10 relays

Fig. 8 Average network bandwidth consumption in the network links of the sites for each media type

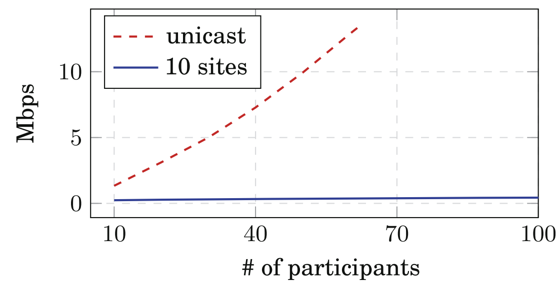


Fig. 9 Average network bandwidth consumption in the network links of the sites when using unicast and the relay mesh

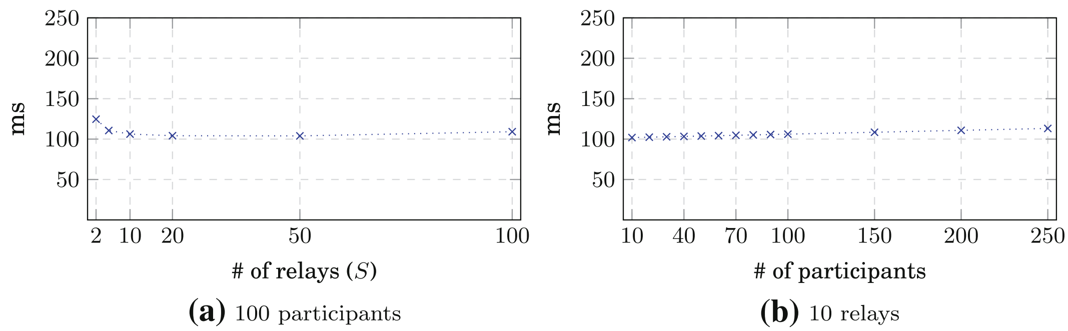


Fig. 10 Time required for the self-healing technique

in Fig. 9. The unicast traffic grows exponentially with the number of participants in spite of using floor control compared to the slight growth when using the relay mesh.

Figure 10 depicts the time required for applying the self-healing technique, which is triggered when an active relay goes down so the participants served by the relay are redirected. This encompasses the detection of the failure, the modification of the relay mesh, the selection of a new relay for each participant, and the subsequent SIP dialogs between the RP and the participants. Figure 10a plots this time for an activity with 100 participants and varying the number of relays in the overlay, while Fig. 10b plots the time for an activity with 10 relays and a different number of participants. The time required to stabilize the overlay after a relay failure rapidly decreases with dispersion for activities with a low number of relays as seen in Fig. 10a, since fewer participants have to be redirected when increasing the number of relays. However, this time increases for highly dispersed activities, as many relays have to be informed when reconfiguring the relay mesh. On the other hand, the stabilization time is directly proportional to the number of participants in the activity as shown in Fig. 10b.

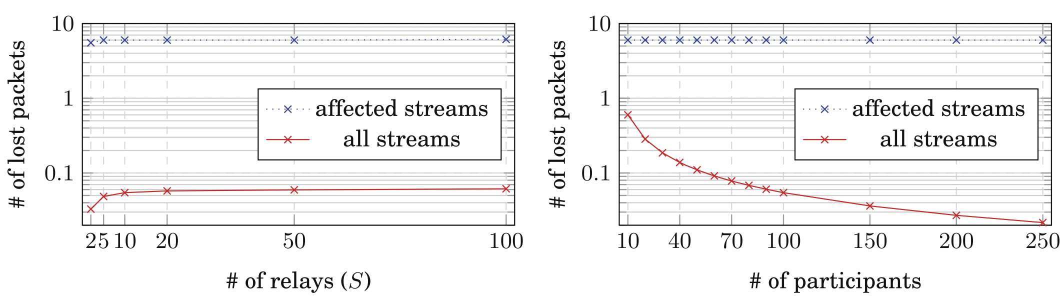
The failure of a relay has some impact on user experience. The failure prevents participants of the site of the relay from receiving and sending data from and to the rest of the activity until the self-healing technique is completed successfully, which leads to a burst of lost packets. Lost packets in continuous media streams produce disturbances, such as audio glitches and video artifacts, which decrease the quality perceived by users. Thus, the quality impact can be measured by means of the number of consecutive lost packets during the re-direction process. On the other hand, packet loss is unacceptable in non-continuous media streams such as annotations, so forward error correction techniques must be used. In this case, the amount of extra information in data packets depends on the maximum packet loss burst to cope with.

Some quality decrease of the video and telepointer streams can be assumed, since video streams only convey the participants' talking heads and interpolation techniques can be used to obtain smooth movements of the telepointers on the shared whiteboard. Conversely audio streams are critical, especially that coming from the instructor, so the impact of packet loss on quality may be significant. This impact depends on whether the packets lost contain silence or data from talk spurts.

Figure 11 shows the average packet loss observed in the received audio streams when applying the self-healing technique. The average number of consecutive lost packets of the streams affected by packet loss is depicted by a blue dotted line. These streams are those received at or sent from the site of the failing relay. The average number of consecutive lost packets observed in all the streams is depicted by a solid red line. Figure 11a plots packet loss varying the number of relays in the overlay, while Fig. 11b plots packet loss as a function of the number of participants.

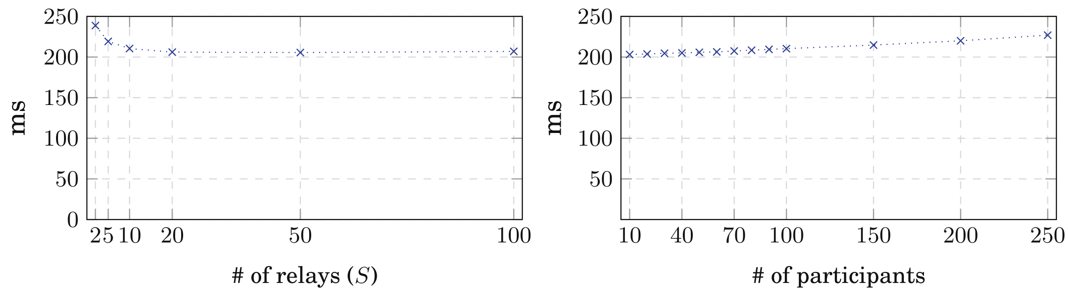
Participants are likely to still receive most of the data streams through IP multicast in activities with a low number of relays when a relay failure occurs, since many participants are co-located. These streams are not affected by packet loss. In the case of the streams with packet loss, the average number of consecutive lost packets slightly increases with dispersion, as the time required to stabilize the relay mesh after a relay failure also increases with dispersion. However, when the dispersion of participants grows, most of the streams received by participants come from the relay mesh, so the likelihood that a stream comes from the site of a failing relay is higher, so packet loss increases. When increasing the number of participants, the number of streams flowing throughout the platform also increases, so the failure of a relay has less influence on the total average of packets lost per stream.

Finally, the time for applying the self-optimization technique is also analyzed in Fig. 12. This is triggered when the failing relay is up again to undo the redirection of participants. The time is plotted as a function of the number of relays in the overlay in Fig. 12a, and the number of participants in Fig. 12b. This time shows a similar trend to the time required for the self-healing technique and depends on the dispersion and number of participants. However, the former is significantly higher than the time required to apply the self-healing technique, as it includes the time required for the registration of the relay with the RP before the redirection process can be undone.



(a) 100 participants **(b)** 10 relays

Fig. 11 Average number of audio packets lost during the redirection process



(a) 100 participants **(b)** 10 relays

Fig. 12 Time required for the self-optimization technique

No packet loss occurs during the self-optimization technique, as the previous relay keeps forwarding data to a participant until he or she is re-directed to another relay. Occasionally, the participant may receive duplicate RTP packets. This is not an issue, since RTP packets contain a sequence number so duplicates can be discarded.

8 Conclusions

In this paper an autonomic platform for synchronous e-training in dispersed organizations is proposed. The platform uses standard protocols to deploy a mesh of RTP relays between geographically dispersed sites of the organization. Properties such as self-deployment, self-organization, self-healing and self-optimization are implemented to provide an efficient multimedia data delivery service between trainees located at their workplaces. Three virtual networks are established during the operation of the platform: a relay mesh to deliver data, a signaling network between participants and the RP, and a mesh control network between the relays and the RP to manage the organization of the mesh.

The use of standard protocols such as SIP for the session signaling and RTP for the data transport makes the platform inter-operable with other software and hardware conferencing solutions. Furthermore, the extensibility of the platform is guaranteed, and its modularity enables the portability of the platform to other data communication approaches without major changes.

The relay mesh used to transport multimedia data makes use of the available network resources efficiently. IP multicast is used where available, so participants communicate in real-time, using low network resources.

Moreover, the self-properties implemented in the platform make the management of the relay mesh automatic and transparent to the user, as well as robust in case of relay failures. The reorganization of the relay mesh is based on the joining and leaving of participants and relay failures.

The main drawback of the platform is its dependency on the RP. The RP is a centralized entity acting as a SIP focus, a SIP registrar and the manager of the relay mesh. A failure in the RP would prevent participants from joining synchronous e-training activities and changes in the organization of the relay mesh, but ongoing activities can continue. This issue can be alleviated using fault-tolerant configurations of the RP.

Acknowledgments This research has been partially funded by the Spanish National Plan of Research, Development and Innovation under the project TIN2011-24903. Financial support (grant:UNOV-10-BECDOC-S) given by the University of Oviedo-Banco de Santander is also acknowledged.

References

1. Granda, J.C., García, D.F., Suárez, F.J., Peteira, I., Uría, C.: A multimedia tool for synchronous distance e-training of employees in geographically dispersed industries. In: Proceedings of the IASTED International Conference on Web-Based Education, pp. 1–8, Innsbruck, Austria (2008)
2. Diot, C., Dabbous, W., Crowcroft, J.: Multipoint communication: a survey of protocols, functions, and mechanisms. IEEE J. Sel. Areas Commun. **15**, 277–290 (1997)
3. Kompella, V., Pasquale, J., Polyzos, G.: Two distributed algorithms for multicasting multimedia information. In: Proceedings of the IEEE International Conference on Communications, pp. 343–349, Geneva, Italy (1993)
4. Zhang, X., Li, X.: Native multicast based videoconference service on CERNET. In: Proceedings of the International Conference on Communication Technology, pp. 1–4, Guilin, China (2006)
5. Diot, C., Neil, B., Bryan, L., Balensiefen, K.D.: Deployment issues for the IP multicast service and architecture. IEEE Netw. **14**, 78–88 (2000)
6. Yamamoto, L., Leduc, G.: Autonomous reflectors over active networks. In: Proceedings of the Symposium on Software Mobility and Adaptive Behaviour, pp. 125–146, York, UK (2001)
7. Nuño, P., Granda, J.C., Suárez, F.J., García, D.F.: Self-* in multimedia communication overlays. Comput. Commun. **36**, 817–833 (2013)
8. El-Sayed, A., Roca, V., Mathy, L.: A survey of proposals for an alternative group communication service. IEEE Netw. **17**, 46–51 (2003)
9. Yeo, C., Lee, B., Er, M.: A survey of application level multicast techniques. Comput. Commun. **27**, 1547–1568 (2004)
10. Čičić, T., Bryhni, H.: Multicast-unicast reflector. In: Proceedings of the Norsk Informatikk Konferanse, pp. 103–114, Kristiansand, Norway (1998)
11. Čičić, T., Bryhni, H., Sølie, S.: Unicast extensions to IP multicast. In: Proceedings of the International Conference on Protocols for Multimedia Systems, pp. 60–69, Krakow, Poland (2000)
12. Hladká, E., Holub, P., Denemark, J.: Teleconferencing support for small groups. In: Proceedings of the TERENA Networking Conference, pp. 1–7, Limerick, Ireland (2002)
13. Hladká, E., Holub, P., Denemark, J.: User-empowered programmable network support for collaborative environment. In: Proceedings of the European Conference on Universal Multiservice Networks, pp. 367–376, Porto, Portugal (2004)
14. Patrikakis, C.Z., Despotopoulos, Y., Robotis, A.M., Lambiris, A.L.: PERIPHLEX: multicast delivery using core unicast distribution with peripheral multicast reflectors. In: Proceedings of the International World Wide Web Conference, p. 292, Budapest, Hungary (2003)
15. Lambrinos, L., Demetriou, E.: An adaptive live media streaming architecture. In: Proceedings of the IARIA International Conference on Advances in Multimedia, pp. 74–77, Athens, Greece (2010)
16. Maraviglia, G., Masi, M., Merlo, V., Licandro, F., Russo, A., Schembra, G.: Synchronous multipoint e-learning realized on an intelligent software-router platform over unicast networks: design and performance issues. In: Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, pp. 1172–1179, Patras, Greece (2007)
17. Zhang, X., Li, C., Li, X.: Multi-party videoconferencing based on hybrid multicast with peer-forwarding. In: Proceedings of the IEEE International Conference on Parallel and Distributed Systems, pp. 812–816, Shanghai, China (2010)
18. Tan, S., Waters, G., Crawford, J.: A performance comparison of self-organising application layer multicast overlay construction techniques. Comput. Commun. **29**, 2322–2347 (2006)
19. Pendarakis, D., Shi, S., Verma, D., Waldvogel, M.: ALMI: an application level multicast infrastructure. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems, pp. 49–60, San Francisco, USA (2001)

20. Liu, Z., He, G., Liu, Z.: Metrino: an application-level multicast system for real-time video communication. In: Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1265–1270, Wuham, China (2005)
21. Liu, Y., Guo, Y., Liang, C.: A survey on peer-to-peer video streaming systems. Peer-to-Peer Netw. Appl. **1**, 18–28 (2008)
22. Jannotti, J., Gifford, D., Johnson, K.L., Kaashoek, M.F., O'Toole Jr, J.W.: Overcast: reliable multicasting with an overlay network. In: Proceedings of the USENIX Symposium on Operating System Design & Implementation, 197–212, San Diego, USA (2000)
23. Chu, Y.H., Rao, S., Zhang, H.: A case for end system multicast. In: Proceedings of the ACM Sigmetrics, pp. 1–12, Santa Clara, USA (2000)
24. Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable application layer multicast. Comput. Commun. Rev. **32**, 205–217 (2002)
25. Castro, M., Druschel, P., Kermarrec, A.M., Nandi, A., Rowstron, A., Singh, A.: SplitStream: high-bandwidth multicast in cooperative environments. ACM Oper. Syst. Rev. **37**, 298–313 (2003)
26. Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. IEEE Commun. Surv. Tutor. **7**, 72–93 (2005)
27. Qiu, T., Chen, G., Ye, M., Chan, E., Zhao, B.Y.: Towards location-aware topology in both unstructured and structured P2P systems. In: Proceedings of the International Conference on Parallel Processing, p. 30, Xian, China (2007)
28. Tran, D.A., Hua, K., Member, S., Do, T.: A peer-to-peer architecture for media streaming. IEEE J. Sel. Areas Commun. **22**, 121–133 (2004)
29. Yin, H., Lin, C., Zhang, Q., Chen, Z., Wu, D.: TrustStream: a secure and scalable architecture for large-scale internet media streaming. IEEE Trans. Circuits Syst. Video Technol. **18**, 1692–1702 (2008)
30. Zhang, X., Liu, J., Li, B., Peter-Yum, T.S.: CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. In: Proceedings of the IEEE International Conference on Computer Communications, pp. 2102–2111, Miami, USA (2005)
31. Venkataraman, V., Yoshida, K., Francis, P.: Chunkyspread: heterogeneous unstructured tree-based peer-to-peer multicast. In: Proceedings of the IEEE International Conference on Network Protocols, pp. 2–11, Santa Barbara, USA (2006)
32. Pianese, F., Perino, D., Keller, J., Biersack, E.: PULSE: an adaptive, incentive-based, unstructured P2P live streaming system. IEEE Trans. Multimed. **9**, 1645–1660 (2007)
33. Liu, Z., Shen, Y., Ross, K.W., Panwar, S.S., Wang, Y.: LayerP2P: using layered video chunks in P2P live streaming. IEEE Trans. Multimed. **11**, 1340–1352 (2009)
34. Huang, N.F., Tzang, Y.J., Chang, H.Y., Ho, C.W.: Enhancing P2P overlay network architecture for live multimedia streaming. Inf. Sci. **180**, 3210–3231 (2010)
35. Wang, W., Chen, Y.: SmartPeerCast: a smart QoS driven P2P live streaming framework. Multimed. Tools Appl. **54**, 445–471 (2011)
36. Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.: RTP: a transport protocol for real-time applications. IETF RFC 3550, http://www.ietf.org/rfc/rfc3550.txt (2003). Accessed 23 Sept 2013
37. Granda, J.C., Nuño, P., Suárez, F.J., Pérez, M.A.: E-pSyLon: a synchronous e-learning platform for staff training in large corporations. Multimed. Tools Appl. **66**, 431–463 (2012)
38. Granda, J.C., García, D.F., Nuño, P., Suárez, F.J.: An efficient networking technique for synchronous e-learning platforms in corporate environments. Comput. Commun. **33**, 1752–1766 (2010)
39. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. IETF RFC 3261. http://www.ietf.org/rfc/rfc3261.txt (2002). Accessed 23 Sept 2013
40. Handley, M., Jacobson, V., Perkins, C.: SDP: session description protocol. IETF RFC 4566. http://www.ietf.org/rfc/rfc4566.txt (2006). Accessed 23 Sept 2013
41. Rosenberg, J.: A framework for conferencing with the Session Initiation Protocol (SIP). IETF RFC 4353. http://www.ietf.org/rfc/rfc4353.txt (2006). Accessed 23 Sept 2013
42. Avižienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secur. Comput. **1**, 11–33 (2004)
43. Elling, R., Pramanick, I., Mauro, J., Bryson, W., Tang, D.: Analytical reliability, availability, and serviceability benchmarks. In: Kanoun, K., Spainhow, L. (eds.) Dependability benchmarking for computer systems, pp. 23–33. Willey, New York (2008)

44. Granda, J.C., Nuño, P., García, D.F., Suárez, F.J.: Security issues in a synchronous e-training platform. In: Proceedings of the IEEE International Conference on Availability, Reliability and Security, pp. 485–492, Vienna, Austria (2011)

Author Biographies

Juan C. Granda received his Ph.D. in Computer Science from the University of Oviedo, Spain, in 2008. He joined the Department of Computer Science and Engineering, University of Oviedo, where he is an Associate Professor. He has been involved in the research activities of the Computer Engineering Area, focusing on performance analysis of synchronous e-learning systems.

Pelayo Nuño received his M.S. degree in Computer Science from the University of Oviedo in 2009, where he is currently working on his Ph.D. degree. He has been involved in several research activities of the Computer Engineering Area. His research interests include multimedia networking and autonomic computing.

Daniel F. García is a full professor in the Department of Computer Science and Engineering at the University of Oviedo, Spain, where he leads the Computer Engineering Area. His current research interest is in the area of self-adaptive multimedia and real-time systems. During the last 20 years, he has led many research projects and coauthored 50 articles in journals and more than 130 papers in conferences and workshops. He is a member of the IEEE Computer Society.

Francisco J. Suárez is a professor in the Department of Computer Science and Engineering at the University of Oviedo, where he received his Ph.D. in 1998. His current research focuses on evaluation of networked multimedia systems and he has been working on several related projects as head of the Interactive Multimedia Services research team.

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.

2011 Sixth International Conference on Availability, Reliability and Security

Security issues in a Synchronous e-Training Platform

Juan C. Granda, Pelayo Nuño, Daniel F. García, and Francisco J. Suárez
Department of Computing Science, University of Oviedo
Campus de Viesques, Gijón 33204, Spain
Email: {jcgranda,nunopelayo,dfgarcia,fjsuarez}@uniovi.es

***Abstract*—Synchronous e-training is emerging as an alternative for developing human resources training plans in large organizations. Real-time communications are used to emulate face-to-face interaction that occurs in on-campus learning environments. However, the security concerns that a synchronous e-training platform must face may compromise the integrity, availability and confidentiality of corporate information, which may lead to serious economic and legal consequences. The disclosure of corporate information or the unauthorized participation in e-training activities must be prevented. In this paper, the security issues in synchronous e-training are identified, and the threats to a real e-training platform are analyzed. The platform is organized into four virtual networks with different security requirements and vulnerabilities. The platform assumes that multicast communications are available in the underlying corporate network. The threats affecting each element of the platform and their impact on e-training activities are discussed. Finally, a security scheme is proposed fixing the aforementioned vulnerabilities. Digital certificates and encryption algorithms solve most of the vulnerabilities, but other techniques such as access control lists and user skills on security basics are essential. Most of the proposed scheme is applicable to other real-time communication systems, since the e-training platform is built using standard technologies commonly used in voice over IP systems.**

***Keywords*-synchronous e-training; security; confidentiality; integrity; availability;**

I. Introduction

Market competitiveness is critical for enterprises in a globalized World. The knowledge and know-how of the organization must be shared among all the participants in the manufacturing and management processes, so human resources training has become a key focus of all organizations.

Internet has emerged as a flexible means to share knowledge, providing new training methodologies supported by multimedia technologies that can be used both in the academic and professional fields. The term e-training is used to refer to the e-learning applied to human resources training. Although asynchronous e-training in the form of self-paced courses is the most commonly used e-training methodology, synchronous e-training provides some benefits that must be taken into consideration, such as a higher level of student engagement due to real-time interactions [1].

Synchronous e-training is especially suited for dispersed learners, reaching them at their workplaces, which reduces displacement costs and time, and increases productivity. Employees can participate in synchronous e-training activities from distant sites of the organization. Synchronous features such as video and audioconferences, shared whiteboards, instant messaging or telepointers, translate face-to-face interactions that occur in the traditional on-campus learning to an online environment. Thus, the presence of the trainer and the rest of the participants in the training process is more notorious than in asynchronous e-training activities. Furthermore, the interactive nature of synchronous e-training makes the training process flexible and effective.

However, synchronous e-training poses some security and privacy challenges. A vulnerable synchronous e-training platform may cause a leak of information with serious economic or legal consequences for the organization. Confidential data is usually interchanged between participants in a training activity. This data includes management and manufacturing practices of the organization that must be kept private. Synchronous e-training platforms must cope with various security issues, taking into account possible malicious actions from external agents.

This paper identifies the security issues associated with synchronous e-training and those presented in a real synchronous e-training platform. The vulnerabilities of the platform are exposed and possible solutions are proposed to achieve a secure platform.

The remainder of this paper is organized as follows. Security principles are introduced in Section II. The architecture of a synchronous e-training platform is detailed in Section III. In Section IV, the security concerns faced by the synchronous e-training platform are enumerated, while Section V exposes the vulnerabilities of the platform. Related work on secure real-time multimedia communication systems is presented in Section VI. The security scheme for the synchronous e-training platform is proposed in Section VII. Finally, Section VIII contains the concluding remarks and outlines future work.

II. Security principles

Information security implies the protection of information from unauthorized access, modification, disclosure and other risks. A secure system satisfies all the information security components. These components have been commonly referred to as the CIA triad [2], standing for confidentiality, integrity and availability. They represent a model for security policy development and help in identifying system vulnerabilities and solutions to guarantee information security.

978-0-7695-4485-4/11 $26.00 © 2011 IEEE
DOI 10.1109/ARES.2011.75

Crytographic techniques have been used to ensure information security [3]. Encryption algorithms guarantee information confidentiality, as the decryption of data is computationally infeasible for an unauthorized entity. Two cryptographic principles are used by encryption algorithms: public-key encryption and symmetric-key encryption. Public-key encryption is based on a couple of keys, public and private key respectively, while symmetric-key encryption is based in a single secret key (or two, but one can be derived from the other).

Digital signatures are fundamental cryptographic primitives for authentication, authorization and non-repudiation. They are based on public-key cryptography. Key distribution is one of the major issues in cryptographic systems. An entity must obtain the public key of another entity in a reliable way in order to establish a communication. This is achieved by using public-key certificates issued by trusted third parties known as Certification Authorities (CAs). X.509 is a standard for issuing certificates.

Various cryptographic algorithms are applied within network protocols for securing network communications. The IPsec protocol ensures confidentiality and integrity of the communication at the network layer. It deploys a virtual tunnel between two entities, encrypting IP packets and encapsulating them into new IP packets. There are also protocols that secure communications at the application layer. TLS 1.1/2/3 and their predecessor SSL provide security services over TCP. DTLS is a version of TLS oriented to securing UDP communications. Finally, Secure RTP (SRTP) provides a secure framework for the exchange of RTP traffic. SRTP requires an appropriate key management procedure for establishing a secure SRTP cryptographic context. This is accomplished by the MIKEY protocol. MIKEY is a key management scheme with low latency, suitable for real-time communications over heterogeneous networks and small interactive groups.

A. *Security in synchronous e-training*

Information confidentiality in an e-training activity must be applied to both the information being shared and the atcivity control data. On one hand, the didactic contents of the training process must be protected. These contents may have great importance for the organization, and its disclosure could cause significant economic damage. On the other hand, the control data of the e-training activity must also be protected. This data includes information about participating users, their location, the privileges of each one during the activity, the architecture of the multimedia data delivery service, the network configuration, and so on. Thus, the platform must identify users, and only those authorized must gain access to the e-training activities and the information being exchanged.

A synchronous e-training platform must ensure that information interchanged between users is not altered, achieving information integrity. The platform must avoid the modification of comments, questions or didactic contents generated by users, disrupting the development of a synchronous e-training activity. This poses a real true challenge, as synchronous e-training platforms are distributed systems composed by many elements communicating concurrently in real-time. Data flowing through the platform might be intercepted and altered during its transmission.

As previously commented, the didactic material used during an e-training activity may contain confidential data from the organization, so it may be desirable to prevent subsequent copies of the material outside the activity to avoid data dissemination. Thus, the documents shared between participants during the e-training activity may be disposed at the end of the activity.

III. A SYNCHRONOUS E-TRAINING PLATFORM

A self-organizing and self-healing synchronous e-training platform is proposed in [4]. This platform provides an efficient multimedia delivery service by automatically organizing itself based on the joining and leaving of participants to an e-training activity. The platform assumes that IP multicast is available in each site of the organization, so the platform interconnects many multicast islands during the e-training activity to deliver multimedia contents to all the participants. This design has proved to be highly efficient [5].

A. *Architecture*

As shown in Fig. 1, various elements are involved during a synchronous e-training activity: a Rendezvous Point (RP), a Binary Floor Control Protocol (BFCP) server, none or several Real-time Transport Protocol (RTP) relays servers and the participants. The RP acts as a Session Initiation Protocol (SIP) focus and a SIP registrar server, providing a standard mechanism to gain access to the activity by authorized users using SIP. The BFCP server is in charge of the floor control policy. The participants and this server communicate using BFCP. The RTP relays forward traffic between different multicast islands, since all data interchanged by participants is transmitted using RTP. Finally, the participants interact with all the aforementioned elements of the platform during the e-training activity.

The e-training platform is organized into four virtual networks: the relay mesh, the signaling network, the mesh control network, and the floor control network.

The relay mesh connects all the RTP relays and the participants in the e-training activity to deliver multimedia content efficiently. A relay is deployed in each multicast island taking part in the activity. Data generated by participants is transmitted using IP multicast and the relay of each multicast island is responsible for forwarding traffic between the island and the rest of the activity. The relays are connected in a full-mesh topology using the Internet as shown in Fig. 1 with thick dashed lines, so data is interchanged among them using unicast connections, in contrast to the IP multicast delivery within each multicast island.

The signaling network connects the RP with the participants, which are geographically dispersed in different sites of the organization. The SIP protocol is used to establish and tear down the multimedia RTP sessions associated with an e-training activity. These sessions are described using the Session Description Protocol (SDP). Every participant establishes

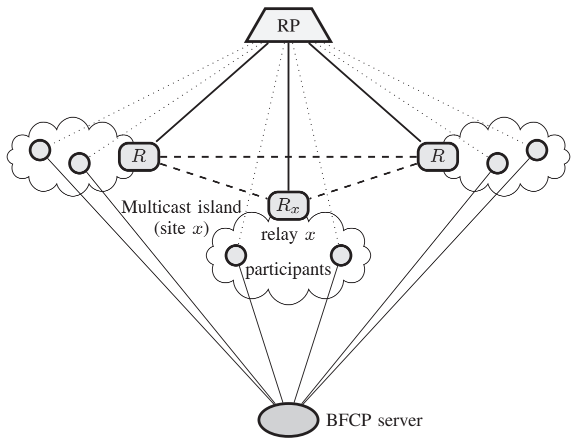


Fig. 1. Architecture of the e-training platform

a SIP dialog with the RP, which plays the role of the focus of a tightly coupled conferencing service. This is illustrated in Fig. 1 with thin dotted lines. Thus, the SIP standard handshake process is used by participants to join e-training activities. Furthermore, the RP acts as a SIP registrar, so participants may register with it. This allows for dial-out scenarios where the RP invites participants to join the e-training activity.

The mesh control network is established between the RTP relays and the RP using TCP connections (solid thick lines in Fig. 1). Although the relays always communicate in a full-mesh topology to keep network latency low, the relays composing the mesh may vary along the e-training activity. The RP is responsible for the reorganization of the relay mesh as participants join and leave. The relays must contact the RP as soon as they start, so the RP is aware of the different multicast islands. The RP maintains a data structure which associates RTP relays with participants within their multicast islands. Thus, the RP can change the relay mesh organization including a new relay when the first participant of its multicast island joins, or excluding an active relay when the last participant of its multicast island leaves. These changes are reported to all the relays in the activity, so they are informed about their peer relays in the mesh.

Finally, the floor control network relates the participants, both the trainer and the trainees, with the BFCP server. This server is responsible for developing the floor control policy of the e-training activity, granting and revoking floors as instructed by the trainer. All the participants establish a TCP connection with the BFCP server for the exchange of BFCP messages as shown in Fig. 1 with solid thin lines.

IV. Security concerns of the platform

Once the architecture of the synchronous e-training platform has been exposed, concerns affecting the information confidentiality, integrity, and availability of the platform are analyzed. Table I summarizes the security concerns of the platform and their compromise on the CIA triad components.

TABLE I
SECURITY CONCERNS IN THE SYNCHRONOUS E-TRAINING PLATFORM

Security concern	Confidentiality	Integrity	Availability
Denial of Service	□	□	■
Degradation of Service	□	□	■
Eavesdropping	■	□	□
Improper user behavior	■	■	□
Alteration of data streams	□	■	□
Participant impersonation	■	■	□
Relay impersonation	■	■	■
BFCP server impersonation	□	□	■
RP impersonation	■	■	■
Activity data manipulation	■	■	■

A denial-of-service (DoS) attack prevents the platform from carrying out synchronous e-training activities normally, compromising the platform availability. Two attack vectors may be used to disrupt the platform. First, an attack to one of the components of the platform (RTP relays, the RP or the BFCP server) may cause a failure of the platform, which prevents many users from participating in the e-training activity. Second, an attact to a participant may prevent a single user, the trainer or a trainee, from participating in the activity.

A degradation-of-service attack is a specific DoS attack, although its main purpose is to degradate the performance of the e-training platform rather than preventing the participation in e-training activities. This attack degrades the learning experience of the participants in the activities.

Eavesdropping is a commonly used term in telephony that is also applicable to synchronous e-training activities. An activity is secretly monitored by an attacker to gain access to the contents being shared and the information generated by participants. The use of several multicast islands during an e-training activity must be taken into account when considering security threats, as multicast traffic is accessible for all the endpoints of a multicast island.

An improper user behavior may cause a severe disruption of synchronous e-training activities. The same user credentials may be shared among many users, which may lead to confusing user identities and unauthorized accesses. A user may also disseminate confidential information. This may be carried out by disgruntled employees or by any trusted participant unconsciously while being spied on by over-the-shoulder eavesdroppers. The attacks from inside organizations are less usual than external attacks, but the former have a greater level of success, being more dangerous and difficult to detect [6].

The alteration of data streams modifies the information interchanged between participants. The original data generated by a participant differs from that arriving at the rest of participants. Documents being shared and the comments from the trainer are critical for the understanding of trainees. Similarly, comments and questions from trainees, as well as the information generated collaboratively may be disrupted by the alteration of one or multiple data streams.

The impersonation of a participant allows an external entity to participate in synchronous activities. The attacker appears to the rest of the platform as an authorized participant by spoofing the participant's identity. This is especially harmful when the trainer is impersonated. The impersonator may give incorrect information to the participants making the training process impossible, or may recommend the trainees erroneous practices deliberately.

On the other hand, the impersonation of an RTP relay may affect the confidentially, integrity and availability of the platform. An attacker may register with the RP as an RTP relay within a ficticious multicast island or impersonate a valid RTP relay. In either case, the attacker may gain access to all the information exchanged between participants. This allows the attacker to change this information or even stop forwarding it to participants within the multicast island, excluding them from the activity.

The BFCP server can be also impersonated. This server grants and revokes floors during e-training activities, so its impersonation may cause a severe disruption. The attacker may revoke all floors, so the interaction between participants becomes impossible.

The impersonation of the RP is another security concern. Since standard SIP dialogs are used for joining and leaving activities, participants joining an activity may be re-directed to a rogue SIP entity acting as the SIP focus of the activity.

Finally, activity data such as the participants and multicast islands (corporate sites) involved, the media sessions and floors available, the duration, etc., is a valuable asset that must be protected. An attacker might gain access to this information and change the multimedia configuration of the whole activity or change the list of authorized users. Currently, this information is stored in the RP as XML documents [7].

V. Vulnerabilities of the Platform

Many risks may compromise the security of the synchronous e-training platform. The vulnerabilities of each of the four virtual networks composing the platform that can be exploited by an attacker must be analyzed.

A. Vulnerabilities of the relay mesh

The relay mesh is the data distribution network. It connects the RTP relays of all the multicast islands and participants using the RTP protocol. Threats to the relay mesh can be broken down into two classes: those related with the RTP protocol and those related with the IP multicast used to deliver data to participants. Fig. 2 illustrates the vulnerabilities of the relay mesh.

Various threats to RTP are detailed in [8] (vulnerability 1). The RTP payload modification implies that an attacker can modify the RTP payload field of the header of the RTP packets using a man-in-the-middle (MitM) attack, which can make data unintelligible. The manipulation of RTP sequence number or timestamps is known as RTP tampering. This attack can make the information unintelligible or even cause a software failure, taking the faulty participant offline. Finally, fraudulent

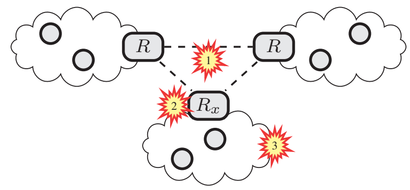


Fig. 2. Vulnerabilities of the relay mesh

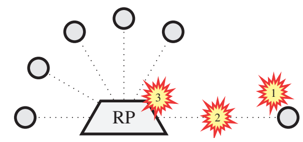


Fig. 3. Vulnerabilities of the signaling network

RTP packets can be injected in the data streams, so the participants may reject original packets [9].

Packet flooding directed to an RTP relay may also compromise the availability of a multicast island or the whole platform (vulnerability 2). A malicious injection of RTP traffic to a relay is propagated by the relay to its peer relays in the mesh, as the relay forwards all incoming traffic to all of them, regardless of the source of the traffic.

On the other hand, several security issues of IP multicast have been reported due to the properties of this kind of communication: open access for receiving from and sending to a multicast group and open group membership [10] (vulnerability 3). Firstly, traditional techniques to restrict the receivers of certain data are not valid in the IP multicast scope as data is delivered to all members of the group. Moreover, any agent can send traffic to a group if the IP address is known, and the receivers lack of a mechanism to verify the authenticity of the source. Furthermore, unauthorized external agents can send fraudulent data to the multicast group with malicious intention such as a packet flooding DoS attack. Finally, open group membership implies that any agent may subscribe a multicast group to receive data delivered to this group, provided that the agent knows the IP address of the group.

B. Vulnerabilities of the signaling network

The signaling network is the virtual network established between the RP and the participants of e-training activities using standard SIP dialogs. Participants should establish a SIP dialog to join and leave an e-training activity. The RP acts as a User Agent Server (UAS), while the participants act as User Agent Clients (UACs). Nevertheless, dial-out scenarios in which the RP initiates the SIP dialog with a previously registered participant are also possible. In this case, the RP acts as a UAC while the participant acts as a UAS. Fig. 3 shows the vulnerabilities of the signaling network.

The threats impacting a SIP based system are discussed in [8]. These threats may affect the e-training platform in different ways.

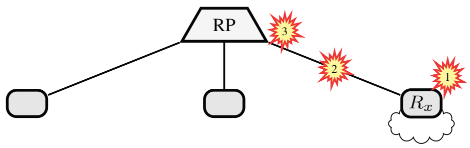


Fig. 4. Vulnerabilities of the mesh control network

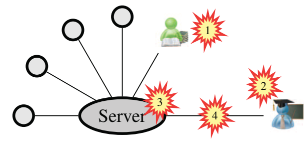


Fig. 5. Vulnerabilities of the floor control network

SIP registration hijacking occurs when the address of a registered User Agent (UA) is replaced by a fraudulent one (vulnerability 1). The RP (acting as a SIP registrar) is accessed by an external agent that changes the UA original address with its address. Thus, if the RP try to establish a SIP dialog with the UA, the e-training activity becomes accessible to the attacking external agent. This gives the attacker access to all data interchanged by participants.

The modification of a SIP message using a MitM attack is undetectable by SIP entities, as SIP messages lack of a built-in integrity mechanism (vulnerability 2). Similarly, intentionally malformed SIP commands could make SIP entities (RP or participants) unstable, as the flexibility and complexity of SIP messages makes their parsing very difficult. A parsing error could lead to a software failure, which may affect the availability of the platform. Besides, SIP Cancel/Bye attacks produce involuntary leavings of participants from the e-training activity. An external agent creates a fraudulent SIP message with the CANCEL or BYE command, terminating an ongoing SIP dialog between the supplanted participant and the RP, which in the end causes the leaving of the participant.

Finally, the attacker could also impersonate the RP intercepting its messages to the participants and becoming the controller of the e-training activity (vulnerability 3). Thus, the attacker may exclude participants or whole multicast islands from the activity, relocate participants, etc.

C. Vulnerabilities of the mesh control network

The mesh control network connects the RTP relays with the RP using TCP connections. This virtual network has various vulnerabilities, as indicated in Fig. 4.

RTP relays register with the RP previously to the begining of a synchronous e-training activity. If an external agent knows the address and port where the RP listens for incoming register requests from relays, it may impersonate a valid relay or identify as a pretended relay (vulnerability 1). Therefore, the attacker will be considered as a valid relay by the RP and it will eventually receive control messages from the RP with information about other RTP relays. Similarly, a MitM attack may be used by an external agent to block registration messages from relays or capture information about the composition of the relay mesh (vulnerability 2). Thus, the relays are not registered with the RP, so the latter cannot include them in the relay mesh of e-training activities.

Once an RTP relay has registered with the RP, the relay closes the TCP connection and waits for the RP for re-establishing the TCP connection and sending control messages to the relay. Then, an attacker may impersonate the RP by establishing a TCP connection with the relay (vulnerability 3). Thus, it may send fraudulent control messages to the relay requesting the relay to join a ficticious relay mesh. The attacker may also include other relays in the relay mesh of an ongoing activity to cause a packet flooding.

D. Vulnerabilities of the floor control network

The floor control network connects the BFCP server with the participants using TCP connections. The BFCP server uses the BFCP protocol to grant and revoke floor requests to participants based on the decissions taken by the trainer of the e-training activity. Fig. 5 shows the vulnerabilities of the floor control network.

An attacker may impersonate a participant by connecting to the BFCP server specifying the identifier of the participant (vulnerabilities 1 and 2). This identifier is an integer, so the attacker only needs to know this identifier and the transport address of the BFCP server. Therefore, the attacker may generate forged floor requests. This is especially harmful when the trainer of the activity is impersonated (vulnerability 2), because the attacker may grant and revoke floors to any participant, which may cause a severe disruption of the activity.

If an attacker impersonates the BFCP server, it may grant or revoke floors to participants, regardless the floor policiy of the activity (vulnerability 3). Eventually, it may grant the audio and video floors to all the participants, producing a packet flooding throught the platform.

BFCP messages lack of a built-in integrity mechanism as occurs with SIP messages. A MitM attack may alter BFCP messages granting or revoking floors to participants different to those planified to be (vulnerability 4). Chair actions from the trainer may also be modified.

VI. Related Work

Building secure platforms based on well-known cryptographic techniques is a current research trend in VoIP, videoconferencing and collaborative learning. There is a wide range of works, from large scale infrastructures to small ad-hoc solutions. The architecture securing AT&T VoIP infrastructure is organized in three levels [11]. The servers providing the VoIP service are located at the top level. The middle level is composed of the network and security elements. The latter detect suspicious interactions between elements of the top and bottom levels. Finally, the bottom level contains the endpoint devices. Signaling is secured using TLS and IPsec, and all the elements exchanging signaling messages are authenticated by digital certificates. Media data is delivered using IPsec and SRTP.

IPsec is commonly used for securing VoIP systems. A security scheme for a SIP-based VoIP system based on IPsec is proposed in [12]. This scheme uses IPsec tunneling between SIP UAs, including specific IPsec-capable firewalls for Virtual Private Networks (VPNs). VPNs are also applied for securing VCNF [13], a videoconferencing peer-to-peer platform, but using SSL instead of IPsec, which relaxes the requirements over the underlying network. VCNF supports symmetric-key encryption for data distribution and public-key encryption for the distribution of the key. The use of VPNs is feasible, as they are usually available in corporate environments. VPNs interconnect different sites of a corporation through public networks, but they are rarely used within the corporate network to secure multimedia communications, so a security scheme exclusively based on VPNs is not possible for the aforementioned e-training platform.

A secure star-topology audio conference platform is described in [14]. The authors propose the use of TLS connections between each participant and the central server, both authenticated using digital certificates. The TLS channel conveys SIP messages, distributes the certificates of all the conference participants and the symmetric key used to cipher data packets, which are transmitted over SRTP.

A procedure for securing end-to-end SIP VoIP communications divided in three steps is suggested in [15]. Firstly, the negotiation of the security mechanism for the subsequent SIP dialog is carried out using TLS. Secondly, the SIP dialog is used to negotiate the call parameters and the key exchange is performed using the MIKEY protocol. Finally, a secure data channel is established using SRTP.

VoIP Defender is presented in [16]. This distributed security architecture detects possible threats and prevents SIP-based VoIP platforms from attacks. SIP messages routed through the platform are analyzed in the background. VoIP Defender deploys elements that analyze SIP traffic using algorithms for detection of attacks, and an decider element for taking countermeasures when an attack is identified.

TLS brings a standard solution to secure SIP communications, so multimedia conferences can be carried out in a secure environment. However, the distributed nature of synchronous e-training platforms imposes additional security challenges that must be faced. Although the communications between every two entities of the platform may be secure, the overall security of the platform cannot be ensured with TLS exclusively.

Security of synchronous e-learning has also gained attention, specially from commercial platforms. WebEx [17] is a Cisco company that provides synchronous e-learning solutions. Each synchronous e-learning session involves two main entities: the Meeting Service Manager (MSM) and the MediaTone Meeting Switch (MMS). The MSM is responsible for the secure multimedia content delivery, while the MMS authorizes the access to the infraestructure deployed for enabling real-time communications. All the participants in the session must be authenticated by the MSM and the MMS using digital certificates and unique cookies for each session. Furthermore, all shared data is protected using proprietary encoding.

TABLE II
CONCERNS FACED BY THE PROPOSED SECURITY SCHEME

Security concern	Mesh	Signaling	Control	Floor
Denial of Service	ACL	ACL	ACL	ACL
Degradation of Service	ACL	ACL	ACL	ACL
Eavesdropping	SRTP/IPsec	(D)TLS	TLS	TLS
Improper user behavior	–	Instruction	–	–
Alteration of data streams	SRTP/IPsec	(D)TLS	TLS	TLS
Participant impersonation				
external agent	SRTP	(D)TLS/X.509	–	TLS/X.509
other participant	*	(D)TLS/X.509	–	TLS/X.509
Relay impersonation	–	–	TLS/X.509	–
BFCP server impersonation	–	–	–	TLS/X.509
RP impersonation	–	(D)TLS/X.509	TLS/X.509	–
Activity data manipulation	–	ACL	ACL	ACL

– Not applicable.
* Open issue.

Netviewer [18] is a platform for web conferencing, desktop sharing, and remote maintenance. It uses two specific servers: the connection server and the communication server. Participants must contact with the connection server to request a session and subsequently contact with the communication server. Communication between participants and the connection server is secured using TLS and digital certificates, while communications with the communication server are encrypted with simmetric-key algorithms. Other commercial platforms such as Adobe Connect [19] and iLinc [20] secure their communications using SSL.

Security schemes from commercial platforms are usually proprietary, so their adaptation to an open e-training platform is not trivial. To the best of our knowledge, our research is the first facing how to ensure the overall security of an open e-training platform.

VII. SECURITY SCHEME

Most of the security threats against the e-training platform exploit the fact that communications are not encrypted. Thus, the first countermeasure to take is to encrypt communications. Communications must be encrypted according to the protocol used to interchange data, so different security solutions are used for each virtual network of the platform. Similarly, since elements within the platform are not authenticated, the impersonation of any of them is possible. Table II summarizes the security concerns faced by the proposed security scheme.

The relay mesh is secured using Access Control Lists (ACLs), SRTP and IPsec. The RP informs an RTP relay when a participant of its multicast island joins and leaves the e-training activity, so the relay maintains an ACL with all the participants authorized to send and receive traffic to and from the activity in the multicast island. Therefore, DoS attacks by packet flooding from external entities injecting multicast traffic can be mitigated. This traffic is not forwarded by the relay to the rest of peer relays.

The use of SRTP at the relay mesh avoids the spoofing of RTP traffic, since only authorized receivers can interpret information. SRTP is an RTP profile that assures RTP data integrity and confidentiality, as it offers services as symmetric-key data encryption and message authentication (integrity protection). Furthermore, the solution assumes that all sites (multicast islands) of the organization are interconnected using VPN channels protected by IPsec. This is very common in corporate networks.

SRTP uses a master key from which session keys are derived to assure data integrity, confidentiality and source authentication in pair-wise communications. However, although the use of SRTP is extensible to multicast scenarios, source authentication is not possible in these scenarios, as session keys used to encrypt and authenticate SRTP packets are shared among all participants. Thus, in those situations where there are many senders, as occurs in the e-training platform, SRTP only assures that a packet has been sent by an authorized entity. SRTP does not prevent the impersonation of an authorized participant by another, for example the impersonation of the trainer by a trainee.

Source authentication is currently an open security issue in multicast SRTP communications with multiple sources. Source authentication could be achieved by using public-key cryptography techniques. However, this is an inefficient approach, as it increases the consumed network bandwidth and has impact in terms of processing time. Public-key cryptography is not feasible for decrypting bulk messages in a real-time scenario, as it may cause an increment in the latency of communications.

At the signaling network, the SIP specification recommends the use of TLS for securing the communication of two entities with SIP over TCP connections. Alternatively, when UDP is the underlying transport protocol, DTLS, that is the equivalent protocol to TLS for datagram transport, can be used for securing communications. The use of SIP with either TCP or UDP depends on the number of participants of e-training activities. If TCP is used, the RP has to keep a TCP connection with every participant in activities, so the scalability of the platform may be compromised with a high number of participants. However, in those activities with a moderate number of participants (from 10 to 30), the use of TCP and TLS is not a scalability issue. The RP accepts both TCP and UDP as transport protocols for SIP communications. Furthermore, X.509 digital certificates are used to authenticate participants and the RP when establishing the TLS or DTLS connections. Only the participants in the ACL of the activity can join it.

X.509 requires a trusted third party certificating that a public key belongs to a specific entity. A CA must be included in the e-training platform, so it issues X.509 certificates for all the entities in the platform enabling the authentication of all parties. This CA can be co-located with the RP, so the RP knows the identity of all entities of the platform.

All the users of the e-training platform must be instructed in best security practices, so as to avoid participants interchanging user credentials or using weak passwords. This instruction must be carried out in advance, as many times the security of the platform is compromised by users' mistakes.

The RP is responsible for creating and distributing the master key used by all the participants in the SRTP sessions of an e-training activity. The SRTP master key distribution is carried out by the RP within the SDP description of the e-training activity using the MIKEY protocol. When a participant establishes a SIP/TLS/TCP or a SIP/DTLS/UDP connection with the RP, the latter provides the participant with the multimedia configuration of the activity and the SRTP master key, so the participant can decrypt and encrypt communications.

The control mesh network can be protected using TLS and X.509 certificates. All the RTP relays must establish a TLS connection in order to register with the RP. The RP keeps a list of authorized RTP relays defined previously to the registration of relays, so the registration of a relay is granted only if the relay is in the list. Once the relay is registered, the TLS connection is closed. The RP must re-establish the TLS connection when creating the control channel with each RTP relay. The relays and the RP are authenticated during TLS setup using X.509 certificates.

At the floor control network, the BFCP standard recommends TLS to get mutual authentication between participants and the floor control server, so TLS can also be used to protect the floor control network. Once a participant have been authorized to join an e-training activity by the RP, the BFCP server is required to include the participant in its ACL and the participant establishes a TLS connection with the BFCP server. Only those participants in the ACL can establish a connection with the BFCP server. Otherwise, the latter refuses the connection. The identities of participants and the BFCP server are verified using X.509 certificates.

Although SIP and BFCP communications can be secured using TLS, the interaction between RP, BFCP server and participants must include additional mechanisms to prevent impersonation. A participant joining an e-training activity poses further risks as illustrated in Figure 6. A trusted participant that is already in the activity might impersonate an incoming participant in the floor control network.

When a new participant joins the activity it receives a BFCP identifier (*id*) from the RP as a result of the SIP dialog establishment. The RP also requests the BFCP server to include the identifier of the incoming participant in the BFCP conference. This identifier is used in the messages interchanged between the participant and the BFCP server and identifies the participant in the BFCP conference. If the attacker tries to impersonate the participant before the BFCP server includes the new identifier in the conference, the attacker will receive an error from the BFCP server indicating that the identifier is unknown. However, if the attack is carried out in the period between the inclusion of the BFCP identifier in the conference and the first contact of the incoming participant with the BFCP server, the impersonation will be successful. The incoming participant will receive an error from the BFCP server indicating that the identifier is already in use.

This vulnerability can be solved by associating the BFCP

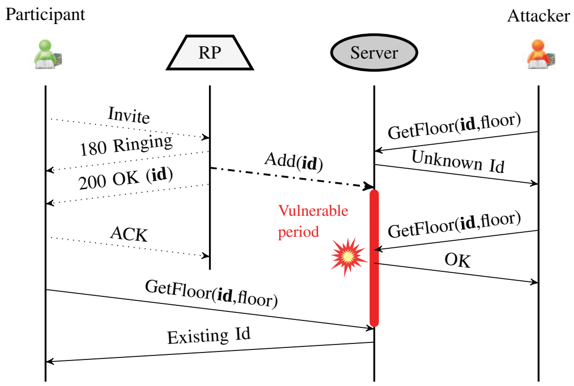


Fig. 6. Vulnerability of the joining process

identifier with the URI of a participant. When the RP requests the BFCP server to include a new BFCP identifier in the conference, it also provides the URI of the participant to which the identifier belongs. Thus, the BFCP server can associate an incoming connection with a BFCP identifier during the TLS negotiation. All the subsequent BFCP messages coming from this connection must refer to this identifier, or otherwise the participant receives an error from the BFCP server.

Finally, users are not allowed to install applications in the computers of the organization in order to avoid the installation of malicious software. It is very feasible that user accounts have limited privileges in corporate environments. Only administrators can install software. Furthermore, corporate firewalls usually block most of the external attacks to elements within the corporate network.

VIII. Conclusions

The security concerns of synchronous e-training platforms have been detailed. A real e-training platform connecting several multicast islands has been analyzed in order to evaluate the threats that have to be faced and the vulnerabilities that can be exploited. This platform uses standard protocols such as SIP, RTP and BFCP. As many of the technologies used in the e-training platform are also used in common VoIP, so do many of the security concerns faced by VoIP are also faced by the platform.

A security scheme has been proposed fixing the majority of the vulnerabilities identified. The proposal takes into consideration the real-time constraints associated with synchronous e-training platforms and it is based on standard solutions and protocols, so it is also extensible for other real-time communication platforms with similar security requirements.

However, the proposed scheme has an open security issue. A technique for source authentication is needed within multicast islands; an authorized participant can impersonate other participants.

Future work will focus on source authentication within multicast islands and the integration of intrusion prevention and detection techniques.

Acknowledgment

This research has been partially funded by the Spanish National Plan of Research, Development and Innovation under the project TIN2008-06045-C02.

Financial support (grant:UNOV-10-BECDOC-S) given by the University of Oviedo-Banco de Santander is acknowledged.

References

[1] C. Taran, "Enabling SMEs to deliver synchronous online training – practical guidelines," *Campus-Wide Inf. Syst.*, vol. 23, no. 3, pp. 182–195, 2006.

[2] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004.

[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 2001.

[4] 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. International Conference on Systems and Networks Communications (ICSNC'10)*, Nice, France, Aug. 2010, pp. 213–220.

[5] J. C. Granda, D. F. García, P. Nuño, and F. J. Suárez, "An efficient networking technique for synchronous e-learning platforms in corporate environments," *Comput. Commun.*, vol. 33, no. 14, pp. 1752–1766, Feb. 2010.

[6] E. E. Schultz, "A framework for understanding and predicting insider attacks," *Comput. & Secur.*, vol. 21, no. 6, pp. 526–531, Oct. 2002.

[7] G. Camarillo, D. Morgan, and J. Urpalainen, "Conference information data model for centralized conferencing (XCON)," Internet Draft, Feb. 2011.

[8] D. Butcher, X. Li, and J. Guo, "Security challenge and defense in VoIP infrastructures," *IEEE Trans. Syst., Man, Cybern.*, vol. 37, no. 6, pp. 1152–1162, Nov. 2007.

[9] M. Adams and M. Kwon, "Vulnerabilities of the Real-Time Transport (RTP) protocol for Voice over IP (VoIP) traffic," in *Proc. IEEE Consumer Communications and Networking Conference (CCNC'09)*, Las Vegas, NV, Jan. 2009, pp. 1–5.

[10] P. Judge and M. Ammar, "Security issues and solutions in multicast content distribution: A survey," *IEEE Netw.*, vol. 17, no. 1, p. 36, Jan. 2003.

[11] W. Marshall, A. F. Faryar, K. Kealy, G. de los Reyes, I. Rosencrantz, R. Rosencrantz, and C. Spielman, "Carrier VoIP security architecture," in *Proc. International Telecommunications Network Strategy and Planning Symposium (Networks'06)*, New Delhi, India, Nov. 2006, article number 4082419.

[12] S. Chacon, D. Benhaddou, and D. Gurkan, "Secure voice over internet protocol (VoIP) using virtual private networks (VPN) and internet protocol security (IPSec)," in *Proc. IEEE Region 5 Conference*, San Antonio, TX, Apr. 2006, pp. 218–222.

[13] C. Du, H. Yin, C. Lin, and Y. Hu, "VCNF: A secure video conferencing system based on P2P technology," in *Proc. IEEE International Conference on High Performance Computing and Communications (HPCC'08)*, Dalian, China, Sep. 2008, pp. 463–469.

[14] A. Treßel and J. Keller, "A system for secure IP telephone conferences," in *Proc. International Symposium on Network Computing and Applications (NCA'06)*, Cambridge, MA, Jul. 2006, pp. 231–234.

[15] J. Kim, S. Yoon, H. Jeong, and Y. Won, "A system for secure IP telephone conferences," in *Proc. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC'08)*, Shanghai, China, Dec. 2008, pp. 356–360.

[16] J. Fiedler, T. Kupka, S. Ehlert, T. Magedanz, and D. Sisalem, "VoIP Defender: Highly scalable SIP-based security architecture," in *Proc. International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm'07)*, New York, NY, Jul. 2007, pp. 11–17.

[17] WebEx Training Center. [Online]. Available: http://www.webex.com/

[18] Netviewer. [Online]. Available: http://www.netviewer.com

[19] Acrobat Connect. [Online]. Available: http://www.adobe.com/products/adobeconnect.html

[20] iLinc. [Online]. Available: http://www.ilinc.com/

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.

2011 IEEE International Symposium on Network Computing and Applications

Towards Resilient Synchronous e-Training Platforms

Juan C. Granda, Pelayo Nuño, Daniel F. García and Francisco J. Suárez
Department of Computer Science and Engineering, University of Oviedo
Gijón 33204, Spain
Email: {jcgranda,nunopelayo,dfgarcia,fjsuarez}@uniovi.es

***Abstract*—Synchronous e-training is emerging as an alternative for developing human resources training plans in large organizations. However, synchronous e-training platforms must face many issues affecting their resilience. Self-recovering and self-protection software techniques should be implemented to enable the continuity of e-training activities faced with network failures or entities of the platform going down. In this paper, the resilience issues in synchronous e-training are identified, and those presented in a real e-training platform are analyzed. The platform includes a self-optimizing technique to make efficient use of network resources as participants join and leave. Furthermore, various self-recovering and a self-protection techniques are proposed, making possible the continuity of ongoing e-training activities with reduced functionality until the automatic full system recovery. Most of the proposed techniques are applicable to other real-time communication systems, since the e-training platform is built using standard technologies such as SIP and RTP, commonly used in voice over IP systems.**

I. Introduction

Market competitiveness is critical for enterprises in a globalized World. The knowledge and know-how of the organization must be shared among all the participants in the manufacturing and management processes, so human resources training has become a key focus of all organizations.

Synchronous e-training is especially suited for dispersed learners, reaching them at their workplaces. Synchronous features such as video and audioconferences, shared whiteboards and applications, instant messaging or telepointers, move face-to-face interactions from the traditional on-campus learning to an online environment.

However, synchronous e-training platforms are very complex systems. A platform usually deploys a multimedia data distribution infrastructure that must cope with heterogeneous network links and end system capabilities, and variable network conditions. Thus, several issues may arise that may affect the overall resilience of the platform, preventing users from participating in synchronous e-training activities.

A resilient synchronous e-training platform should provide mechanisms for facing unforeseen events and automatically recovering from failures in different entities of the platform. Additionally to fault-tolerant hardware solutions based on mirroring of critical entities and hot spare configurations that have been traditionally used, synchronous e-training platforms may implement self-recovering techniques, so e-training activities can continue in case of a failure in any entity of the platform. A combined solution based on fault-tolerant hardware configurations and self-recovering techniques provides a high resilient level for synchronous e-training activities. Nevertheless, in those situations where fault-tolerance hardware configurations are not possible (high costs, many critical entities, etc.), self-recovering techniques achieve a reasonable level or resilience, as e-training activities may continue although with reduced functionality.

This paper identifies the issues affecting the availability, reliability and safety of synchronous e-training activities, and those presented in a real synchronous e-training platform. Various self-recovering and self-optimizing mechanisms are proposed. Synchronous e-training activities can continue when a failure occurs in any of the entities of the platform. These mechanisms are focused on recovering the activity to a fully functional state as soon as the failing entity is running again.

The remainder of this paper is organized as follows. Related work on resilient real-time group communication systems is discussed in Section II. Resilience principles affecting synchronous e-training activities are introduced in Section III. The architecture of a synchronous e-training platform and its resilience concerns are detailed in Section IV. Section V proposes various techniques to improve the overall resilience of the platform. Finally, Section VI contains the concluding remarks and outlines future work.

II. Related Work

A resilient system provides and maintains an acceptable level of service in the face of various faults and challenges to normal operation [1].

Disruption tolerance is the ability of a system to tolerate network connectivity disruptions among its components [2]. Works on this field mainly focus on three research trends. Many works analyze the issues affecting ad-hoc networks, such as mobility in wireless networks [3], or lifetime in energy-constrained networks [4]. Other works focus on large-delay issues, proposing delay-tolerant networking systems [5]. Finally, other authors promote resilient overlay networks (RONs) [6].

A survey of RONs research works is presented in [7], classifying them in three groups: cross-link, in-tree, and multiple-tree redundancy. A resilient node is achieved by using several adjacent nodes and back-up paths among them in cross-link and in-tree redundant techniques. Cross-link redundancy connects random peers [8], while in-tree redundancy organizes peers in clusters [9]. Several data distribution trees are built in multiple-tree redundancy, so nodes composing the overlay network are organized in different ways [10].

978-0-7695-4489-2/11 $26.00 © 2011 IEEE
DOI 10.1109/NCA.2011.42

IEEE computer society

RONs are based on self-recovering and self-optimizing mechanisms that are able to detect failures, recover from them and dynamically discover optimal network paths among nodes [11]. Measures such as packet loss rate, delay jitter, round-trip time or application throughput are parameters frequently used when applying these mechanisms. Self-recovering techniques can also be based on idle times [12].

Research works on RONs are usually focused on peer-to-peer (P2P) and application layer multicast (ALM) techniques. However, overlay networks based on reflectors achieve lower latency on communications, which makes them more suitable to real-time interactive communication systems such as e-training platforms. Thus, this paper focuses on a RON composed of various reflectors which is used to support synchronous e-training activities.

III. Resilience in Synchronous e-Training

Synchronous e-training platforms are complex systems that use the underlying network and intermediate entities to carry out synchronous e-training activities. Many issues may arise affecting the availability, reliability and safety of a platform. It is essential that e-training activities can continue in case of network or entity failures without disturbing the training process. Although a temporary controlled reduction of functionality due to a failing entity in the platform can be assumed, the platform must automatically recover all functionality when the failing entity is running again.

Various techniques can be used to increase the availability, reliability and safety of a synchronous e-training platform.

Firstly, self-recovering techniques can be used to automatically recover from a failure, making possible the continuity of the e-training activities. Furthermore, each recovered entity must return to an operational context similar to its state before failing. For example, when a participant re-joins an activity after temporally losing the connection, the state of the activity (documents being shared, floors held, audio and video streams, etc.) must be recovered.

Secondly, self-optimizing techniques allow for an efficient use of the available network and computing resources, resulting in higher reliability, as the load supported by network links and platform entities is limited. A platform should re-organize itself to overcome the obstacles associated with the changing conditions of the network and participants. Similarly, actions to recover from or minimize the effects of failures usually imply a re-organization of the data distribution paths.

Finally, self-protection techniques can be used to ensure the safety of the platform. Thus, the platform should move to a safe state whenever a failure occurs for e-training activities to be barely disturbed. For example, a self-protection technique must ensure that floor control policies are developed, avoiding anarchic states where activities are unmoderated when a failure occurs in any of the entities involved in the floor control management.

All these techniques depend on the detection of failures in platform entities, so the platform must also provide mechanisms to detect failing entities. However, various specific difficulties must be faced when developing these techniques in synchronous e-training platforms.

The real-time nature of communications among participants is one of the most challenging difficulties that a synchronous e-training platform must face. The data delivery service must be designed to allow low-latency interactive communications, requiring that network resources are not saturated. However, the network resources in a corporate environment are limited, as data of e-training activities must share the available bandwidth with other kinds of traffic necessary for the daily necessities of the organization. Thus, self-optimizing techniques are critical for an efficient use of resources.

Synchronous e-training platforms are complex systems with multiple points of failure, so fault-tolerant hardware solutions are difficult to apply. Many distributed elements operate in the delivery of multimedia data, the admission of participants, the management of floor control, and so on. A failure in a single element of the platform may prevent users from participating in activities. Self-recovering techniques should take this fact into consideration.

Another important challenge is heterogeneity, which is presented at the network level and at the operational level. At the network level, synchronous e-training activities involve many participants located in different sites of an organization. Thus, an e-training platform must cope with different network link available bandwidths and configurations, as well as the different configurations of participants' computers. At the operational level, e-training platforms may deploy a hybrid model combining centralized and distributed group communication delivery models such as P2P, reflectors and ALMs. In that case, the management of the entities is different depending on their function.

IV. A Synchronous e-Training Platform

A self-optimizing and self-recovering platform for synchronous e-training is proposed in [13]. This platform provides an efficient multimedia delivery service by automatically organizing itself based on the joining and leaving of participants to an e-training activity. The platform assumes that IP multicast is available in each site of the organization, so the platform interconnects several multicast islands in an overlay network during the e-training activity to deliver multimedia contents to all the participants. This design has proved to be highly efficient [14].

As shown in Fig. 1, various entities are involved during a synchronous e-training activity: a Rendezvous Point (RP), a Binary Floor Control Protocol (BFCP) server, none or several Real-time Transport Protocol (RTP) relays servers and the participants. The RP acts as a Session Initiation Protocol (SIP) focus and a SIP registrar server, providing a standard mechanism to gain access to the activity by authorized users using SIP. The BFCP server is in charge of the floor control policy. The participants and this server communicate using BFCP. The RTP relays forward traffic between different multicast islands, since all data interchanged by participants is

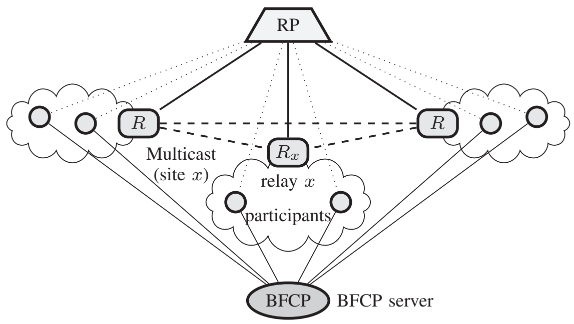


Fig. 1. Architecture of the e-training platform

transmitted using RTP. Finally, the participants interact with all the aforementioned entities of the platform during the activity.

The e-training platform is organized into four virtual networks: the relay mesh, the signaling network, the mesh control network, and the floor control network. The relay mesh connects all the RTP relays and the participants in the e-training activity forming an overlay network to deliver multimedia content efficiently. A relay is deployed in each multicast island taking part in the activity. Data generated by participants is transmitted using IP multicast and the relay of each multicast island is responsible for forwarding traffic between the island and the rest of the activity using unicast connections (thick dashed lines).

The signaling network connects the RP with the participants. The SIP protocol is used to establish and tear down the multimedia RTP sessions associated with an e-training activity. These sessions are described using the Session Description Protocol (SDP). Every participant establishes a SIP dialog with the RP (thin dotted lines), which plays the role of the focus of a tightly coupled conferencing service. Furthermore, the RP acts as a SIP registrar, so participants may register with it.

The mesh control network is established between the RTP relays and the RP using TCP connections (solid thick lines). Although the relays always communicate in a full-mesh topology to keep network latency low, the relays composing the mesh may vary along the e-training activity. The RP is responsible for the re-organization of the relay mesh as participants join and leave. The relays must contact the RP as soon as they start, so the RP is aware of the different multicast islands. The RP maintains a data structure which associates RTP relays with participants within their multicast islands. Thus, the RP can change the relay mesh organization including a new relay when the first participant of its multicast island joins, or excluding an active relay when the last participant of its multicast island leaves. These changes are reported to all the relays in the activity, so they are informed about their peer relays in the mesh.

Finally, the floor control network relates the participants, both the trainer and the trainees, with the BFCP server. This server is responsible for developing the floor control policy of the e-training activity, granting and revoking floors as instructed by the trainer. All the participants establish a TCP connection with the BFCP server for the exchange of BFCP messages (solid thin lines).

A. Resilience Issues Faced by the Platform

The platform provides some mechanisms focused to enhance the resilience of the relay mesh. Although the RP controls activities and authorizes users to join them, it is not responsible for delivering data among users, so an RP failure does not prevent participants from interchanging data. An RP failure prevents new participants from joining the activity and implies that the relay mesh is not re-organized as participants leave.

The platform includes a self-recovering process to ensure the availability of an activity in case of one or multiple relay failures.The platform detects a failure in an active relay when the TCP connection between the RP and the relay is lost. When the connection is unexpectedly closed, the RP assumes that the relay has failed, so the RP re-directs all the participants in the multicast island of the relay to other active relays. Thus, the participants can continue sending and receiving multimedia data to and from the activity. The relays to which participants are re-directed are chosen based on the number of participants that they are already supporting.

The re-direction of participants to other relays is transparent to participants. The RP acting as the SIP focus of the activity sends a SIP re-INVITE message to all the participants to be re-directed, so the multimedia sessions described in the SDP description included in each one of these SIP messages are pointed to another relay. Finally, when the failing relay is running again, the re-direction process is reverted and the participants are re-directed to their original relay.

However, the re-direction process has some drawbacks. Re-directed participants must use unicast delivery to communicate with the relays, so an increase of network resource consumption occurs. Moreover, the algorithm used to select the relay to which participants are re-directed is not optimal. The algorithm only takes into account the number of participants that are already associated with relays.

The BFCP server is usually co-located with the RP in the same computer, although they can also be deployed separately. A TCP connection is established between the RP and the BFCP server. If the latter fails, the floor control is disabled in a controlled manner. This self-protection technique implies that only the trainer is authorized to hold all the floors of the e-training activity, along with those participants that had been granted previously to the BFCP server failure. Thus, floor control policies are already satisfied.

B. Resilience Vulnerabilities of the Platform

Relays are critical entities of the platform, as a failure in one relay compromises the participation of the users within its multicast island in the e-training activity. The use of redundant relays in each multicast island implies very high costs due to the possible high number of multicast islands. On the contrary, the platform assumes that relays may fail, so the RP must apply the aforementioned self-recovering algorithm regularly. However, this algorithm does not consider the conditions of the network or the resources used in each relay, which may lead to erroneous decisions. A participant is re-directed to the relay supporting the lowest number of participants regardless the available bandwidth of the network of its multicast island. The RP should gather information from the relays about the conditions of their network links and the computing resources being used. Thus, the RP can make an optimal decision in the re-direction of participants to relays.

Although activities may survive an RP failure, if the RP becomes up again, it cannot recover the control over the activity, so the activity must be re-started in order to the RP gaining control again. However, it is necessary to go further and provide a technique that allows the RP to recover the control of the activity without stopping it. Thus, new participants can join the activity once the RP is up. A similar situation occurs when the BFCP server fails. The BFCP server should recover the state of all floors in order to grant or revoke floors once it is running again.

Finally, the platform has no mechanism to detect participants leaving the activity ungracefully or failing. This may lead to a waste of resources, as the last participant remaining in a multicast island may have left the activity, and the rest of the relays of the activity keep forwarding data to the multicast island of the absent participant. Therefore, it is necessary to define additional mechanisms to detect participant failures which may lead to a re-organization of the relay mesh.

V. Resilience Scheme

Based on the previously analyzed vulnerabilities, several software mechanisms are proposed to increase the overall resilience of the synchronous e-training platform. First, a proposal for the detection of failing entities in the platform by the RP is described. Next, an improved algorithm for selecting the relay to which participants are re-directed when their original relay fails is formulated. This requires the RP to gather information about network conditions from relays regularly. Finally, a self-recovering technique that allows the RP to gain the control of an ongoing activity after recovering from a critical failure is presented.

A. Failing Entities Detection

The efficient organization of the relay mesh may be compromised due to failures of its entities. A failure of a relay causes the re-direction of all of its associated participants. On the other hand, a failure of a participant which is the last remaining of a multicast island implies a waste of resources, as traffic of the activity is forwarded to a multicast island with no active participants. Thus, it is critical to detect failures of these entities that may cause a re-organization of the relay mesh. Since the RP is responsible for the re-organization of the relay mesh, it must be able to detect failing entities. This is especially complex in the case of participants, as the communications between the RP and participants are not persistent. The RP and the participants interchange SIP messages over UDP, which is not connection oriented, so the RP is not aware of participants leaving ungracefully. To solve this problem, a twofold mechanism for detecting failing entities is proposed: a short-term detection and a long-term detection mechanisms.

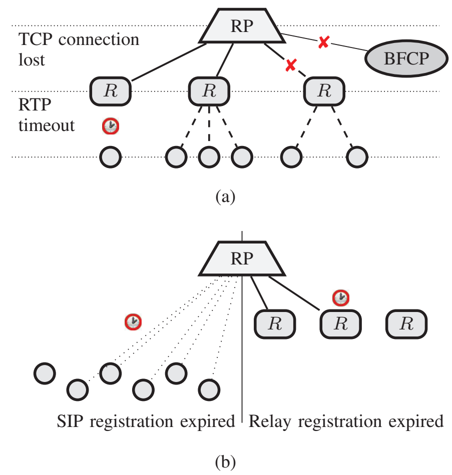


Fig. 2. Failing entities detection tree: (a) short-term; (b) long-term

The short-term detection mechanism tries to detect failing entities as soon as possible. This mechanism is illustrated in Fig. 2a. As can be seen, a three-level detection tree is built. The RP is the root of the tree, the relays and the BFCP server are connected with the root, and the participants are the leaves of the tree. Each entity of the tree is responsible for detecting the failure of its descendants and notifying this fact to its ancestor. Thus, the relays notify the RP when one or several of its associated participants fail or leave the activity ungracefully. This information is used by the RP to re-organize the relay mesh. A relay detects that a participant is not active by snooping RTP traffic. When a relay does not receive the traffic from a participant, it considers that the participant has failed or abandoned the activity. On the other hand, the RP detects failing relays. This is achieved using the TCP connection with each relay as a keep-alive mechanism. A failure of the BFCP server is detected by the RP when the TCP connection between them is unexpectedly closed.

A long-term detection mechanism is also proposed in Fig. 2b. This mechanism detects failures not reported by the short-term detection mechanism. It is based on the registration processes that must be carried out by relays and participants. Relays must register with the RP when they start running, while participants must perform a SIP registration with the RP acting as a SIP registrar. Both registrations expire, so relays and participants should refresh registrations periodically. In this case, the RP expires directly each entity of the platform. If the registration timer expires, the entity is considered as failing by the RP.

B. Re-direction of Participants

Participants in a multicast island must be re-directed to other relays when the relay of the multicast island fails.

An enhanced version of the algorithm to select the relay to which participants are re-directed can be achieved taking into consideration the conditions of the underlying network and the resources being consumed in each relay.

Let Pu_i and Pm_i be the maximum number of unicast participants (re-directed participants) and multicast participants that the relay i can support, respectively. These numbers are provided by the relay when it registers with the RP. Let pu_i and pm_i be the number of unicast and multicast participants currently supported by relay i, respectively. Relay i regularly provides the RP with estimators of the number of lost packets per second (l_i) and the CPU usage (c_i). The l_i estimator provides an indication of the network conditions, as a high loss rate usually implies a degradation of network conditions. On the other hand, the c_i estimator provides an indication of the amount of extra traffic that the relay can forward. Then, Eq. (1) is proposed considering all these variables.

$$\frac{(Pu_i - pu_i) \times (Pm_i - pm_i)}{Pu_i \times Pm_i} \times (1 - l_i) \times (1 - c_i) \qquad (1)$$

When the RP has to re-direct a participant, it computes (1) for each active relay in the mesh, resulting in a number in $[0, 1]$. The RP re-directs the participant to the relay with the highest value.

The first operands in (1), $(Pu_i - pu_i)/Pu_i$ and $(Pm_i - pm_i)/Pm_i$, prevent relay i from supporting more participants than expected. These operands allow the algorithm to balance participants among the relays. When relay i reaches the maximum number of supported participants, either unicast or multicast, it will not be assigned more participants, as the result of computing (1) will be 0.

The operand $(1 - l_i)$ refines the algorithm including an estimator about the conditions of the network link of relay i. Thus, the algorithm does not re-direct participants to relays with degraded network conditions. Let r_i be the packet loss rate per second at relay i, obtained by analyzing incoming RTP streams. Then, l_i can be computed as:

$$l_i = min\left(\frac{r_i}{0.15}, 1\right) \qquad (2)$$

The variable l_i represents a packet loss rate at the relay i from 0% to 15% or grater, but scaled to $[0, 1]$. The value of 15% has been chosen empirically, since such a high packet significantly degrades the quality perceived by users [15]. The variable l_i is assigned a value close to 0 when network conditions at relay i are optimal and there is no packet loss, and with a value close to 1 when the packet loss rate reaches a value of 15%. This prevents the maximum number of supported participants (Up_i and Mp_i) being too optimistic, or the network link of the relay becoming overloaded. Unfortunately, a value of l_i close to 1 may also indicate a temporary network issue not reflecting the real network conditions, so the relay would be discarded to be associated more participants incorrectly.

The operand $(1 - c_i)$ estimates the resources consumed at relay i. The variable c_i is the CPU usage scaled by 1.1 (and truncated to 1) to avoid CPU usage above 90%. This operand prevents the algorithm from re-directing participants to relays with a high load that might not be able to forward traffic to and from this extra participant.

The variables l_i and c_i may vary due to changing conditions in the network and relay i. They can be smoothed considering their moving average, so transient peaks are avoided. These peaks can be caused by temporary network issues or high resource consuming processes running in the relay. Similarly, the rate at which the relays informs the RP about these estimators must be a compromise between bandwidth used and reliability of measurements.

This algorithm enhances the re-direction process with objective criteria concerning to the state of each relay beyond the number of participants supported. Moreover, assuming that network conditions and resources used are optimal for all the relays, the algorithm ensures that participants are re-directed so that each relay has the same percentage of participants re-directed of all that it supports. Thus, the probability that a participant is re-directed to a relay would depend on the maximum number of participants supported by the relay.

```
procedure OngoingActivityRecovery(activity)
    WaitForRelays()
    for all user ∈ Users(activity) do
        relay ← OriginalRelay(user)
        if relay ∈ RegisteredRelays() then
            sdp ← CreateSDP(activity, relay)
            Reinvite(user, sdp)
            Associate(activity, relay, user)
            ActiveRelays(activity) ← relay
        else
            ReDirect(activity, relay, user)
        end if
    end for
end procedure

procedure ReDirect(activity, relay, user)
    newrelay ← RedirectTo(ActiveRelays(activity))
    sdp ← CreateSDP(activity, newrelay)
    Reinvite(user, sdp)
    Associate(activity, newrelay, user)
    MarkAsRedirected(activity, relay, newrelay, user)
end procedure
```

Fig. 3. Ongoing activity control recovery.

C. Recovering Control of Ongoing Activities

As previously commented, the RP is not able to recover the control over an ongoing activity when it is up again after going down. An algorithm for self-recovering from failures of the RP is described in Fig. 3, providing that the state of synchronous e-training activities is continously saved to a persistent database.

The state of an activity is updated whenever a participant joins or leaves, a relay registers with the RP, the relay mesh organization changes, or a floor is granted or revoked. The state of the activity includes data such as the IP addresses of relays and participants, the relationship between participants and relays, and the floors granted to each participant. Data about re-directed participants is also included. This data is stored in XML files.

The recovering of an ongoing activity takes several steps. First, the RP recovers the state of the activity from the XML files, and waits for relay registration requests from those relays that were part of the activity previously to the RP going down. The relays have already detected the RP failure, as the TCP connection with the RP has been lost, so they try to re-register with the RP continuously. The RP starts a timer waiting for each relay to complete the registration process. If the timer expires, those relays that have not registered yet are considered by the RP as failed during the ongoing activity, so the RP exclude them from the relay mesh.

In parallel, the RP re-establishes the SIP dialog with each participant. The RP creates an SDP description of the RTP sessions that composed the activity, and initiates a SIP dialog with each participant using a SIP re-INVITE message. Before sending the message, the RP checks if the relay associated with the participant is already registered. If so, the RP sends the SIP re-INVITE message, sets the participant as recovered, and includes the relay in the relay mesh of the activity. If the SIP dialog cannot be re-established with a participant, the RP considers that the participant has left the activity.

Otherwise, if the associated relay has not registered and the timer has expired, all participants associated with the relay must be re-directed. The RP uses the previously described re-direction algorithm. The RP points the multimedia RTP sessions in the SDP description to the selected relay, sends a SIP re-INVITE message to the participant, and sets the participant as recovered. Additionally, the participant is also set as re-directed, storing its original relay and current relay (re-directed) for reverting the re-direction if the original relay re-registers with the RP.

The BFCP server can recover the control over the floors in an e-training activity in a similar way. The server recovers the state of all the floors of the activity from the persistent database and notifies the RP that it is running again. Then, the RP sends re-INVITE messages to all the participants in the activity so they re-establish the TCP connection with the BFCP server. Therefore, trainees can request or release floors again, as well as the trainer grant or revoke them.

VI. Conclusions

The resilience concerns of synchronous e-training platforms have been detailed and a real e-training platform has been analyzed in order to evaluate the resilience issues that have to be faced. This platform uses standard protocols such as SIP, RTP and BFCP. Most of the issues identified can be applied to other e-training platforms and even to other real-time communication systems such as voice over IP, as many of the technologies used are the same.

A resilience software scheme, which can be used in combination with other hardware or software fault-tolerant mechanisms, has been proposed fixing the majority of the previously analyzed issues. The platform includes a self-optimizing technique for the relay mesh, which is re-organized as participants join and leave e-training activities. Thus, the network resources are used efficiently. Various self-recovering techniques have also been developed to face failures in the relay mesh, the RP or the BFCP server, making possible the continuity of e-training activities with reduced functionality until the automatic full system recover. Furthermore, a self-protection mechanism avoids violations of the floor control policy of activities when the BFCP server fails.

Acknowledgment

This research has been partially funded by the Spanish National Plan of Research, Development and Innovation under the project TIN2008-06045-C02.

Financial support (grant:UNOV-10-BECDOC-S) given by the University of Oviedo-Banco de Santander is acknowledged.

References

[1] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, 2004.

[2] J. P. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, no. 8, pp. 1245–1265, 2010.

[3] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: ubiquitous and robust access control for mobile ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1049–1063, 2004.

[4] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 40–50, 2002.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. ACM SIGCOMM Conf. on Computer Communications*, 2003, pp. 27–34.

[6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM Symp. on Operating Systems Principles (SOSP'01)*, 2001, pp. 131–145.

[7] S. Birrer and F. E. Bustamante, "A comparison of resilient overlay multicast approaches," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1695–1705, 2007.

[8] S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan, "Resilient multicast using overlays," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 237–248, 2006.

[9] S. Birrer and F. E. Bustamante, "Resilient peer-to-peer multicast without the cost," in *Proc. 12th SPIE/ACM Multimedia Computing and Networking Conf. (MMCN'05)*, 2005, pp. 113–120.

[10] E. Setton, P. Baccichet, and B. Giro, "Peer-to-peer live multicast: A video perspective," *Proc. IEEE*, vol. 96, no. 1, pp. 25–38, 2008.

[11] J. Li, P. L. Reiher, and G. J. Popek, "Resilient self-organizing overlay networks for security update delivery," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 189–202, 2004.

[12] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming," in *Proc. IEEE INFOCOM*, 2005, pp. 2102–2111.

[13] 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. Int. Conf. on Systems and Networks Communications (ICSNC'10)*, 2010, pp. 213–220.

[14] J. C. Granda, D. F. García, P. Nuño, and F. J. Suárez, "An efficient networking technique for synchronous e-learning platforms in corporate environments," *Comput. Commun.*, vol. 33, no. 14, pp. 1752–1766, 2010.

[15] A. Raake, *Speech Quality of VoIP: Assessment and Prediction*. West Sussex, UK: John Wiley & Sons, 2006.

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.

Multimed Tools Appl
DOI 10.1007/s11042-012-1061-9

E-pSyLon: a synchronous e-learning platform for staff training in large corporations

Juan C. Granda · Pelayo Nuño · Francisco J. Suárez · María A. Pérez

Abstract Synchronous e-learning is becoming increasingly popular as a tool for developing human resource training plans in large corporations. A synchronous e-training platform has been designed and implemented to explore the possibilities of synchronous e-training in a large corporation. The client tool is used by learners and instructors to collaborate in real-time during e-training activities. It is based on standard protocols also used in voice over IP, which makes it inter-operable with other software and hardware devices. The functionalities offered by the client tool have been chosen based on those provided by commercial tools, but focusing on the specific characteristics of users and network conditions of corporate environments. This facilitates highly interactive synchronous e-training activities. Multimedia configuration of activities is flexible, so participants with different network link capabilities can join activities. The e-training platform has been tested in real e-training activities in ArcelorMittal Spain. The opinions of the users reflect an enhanced learning experience when using the client tool.

Keywords Synchronous e-learning · E-training · Staff training · Educational multimedia tools

J. C. Granda ( ) · P. Nuño · F. J. Suárez
Department of Informatics, University of Oviedo, Campus de Viesques, 33204 Gijón, Spain
e-mail: jcgranda@uniovi.es

P. Nuño
e-mail: nunopelayo@uniovi.es

F. J. Suárez
e-mail: fjsuarez@uniovi.es

M. A. Pérez
La Toba ArcelorMittal Training Center, Avilés, Spain
e-mail: Maria-Aranzazu.Perez@arcelormittal.com

Published online: 14 March 2012

1 Introduction

Globalization has an enormous impact on the manufacturing and management processes of enterprises. New processes and technological advances must be adopted to increase competitiveness, so enterprises must define training plans to improve the skills of their personnel as innovations are introduced.

Generally, multi-national enterprises have a specific department for human resource training, which designs and develops training activities. However, the continuing training process implies high costs, especially when traditional on-campus learning is used. This is due to various factors.

Firstly, employees spend useful working time attending training activities. The total duration of an activity is typically broken down into sessions of a few hours per day. As a result, employees have less time for daily work. Secondly, employees must move from their workplace to the training center in order to participate in training activities. The distance can vary from a few kilometers (if the training center is close to the workplace) to hundreds of kilometers. The displacement time reduces the useful working time of employees still more. In addition, as the company does not usually hire transport services, it is necessary to pay employees to make their own way to the training center. Large multi-national corporations are widely dispersed, so displacement costs are high.

Furthermore, it is often necessary for an expert to travel to different sites of the corporation to teach a highly specialized training activity in order that all sites of the corporation share the same procedures and knowledge. This can be prohibitively expensive when there are many sites, or when the number or employees attending the activity in each site is low.

E-learning has emerged as a viable alternative to traditional on-campus learning, where learners and instructors share the same time and physical space. Information technologies can be used to support the training process without real face-to-face interaction. In fact, learners and instructors may be separated in space and/or time. Therefore, displacement costs in terms of both time and money can be saved when using e-learning tools to develop remote training activities. Distant training is possible when specific equipment is not required during training activities. The term e-training is commonly used to refer to e-learning applied to the training of human resources.

Because of the restrictions imposed on learning processes, e-learning can be of two types: asynchronous or synchronous. Participants in asynchronous e-learning activities can join them from anywhere at any time. On the contrary, participants in a synchronous e-learning activity must join the activity at the same time in order for the learning process to take place.

The utilization of synchronous e-learning has multiple advantages over other instructional strategies. Synchronous e-learning is especially suited for connecting dispersed learners. In the case of staff training, synchronous e-learning is useful for reaching employees at their workplace. Employees can interact directly with the instructor, and the same instructor can teach a specialized e-training activity to all of the sites of the corporation. Synchronous e-learning has many similarities to traditional on-campus learning, since it allows face-to-face interactions in a virtual classroom.

In contrast to asynchronous e-learning, synchronous e-learning enables real-time interactions between learners, which promotes collaborative learning. The sense of immediacy perceived by learners makes them feel comfortable during the learning process and enhances the instructor's presence, so potential anxieties suffered by learners due to an impersonal learning experience are allayed [7].

This paper describes a new synchronous e-training platform tailored to the specific requirements of staff training in large corporations. Its design is highly influenced by employee skills and the typical characteristics of a corporate network. This platform takes advantage of the specific characteristics of corporate networks to carry out synchronous e-training activities using network resources efficiently. Thus, the possible activities using the platform can range from highly interactive activities with a low number of participants to low interactive activities with a huge number of users. The instructor is able to control the interactions among all the participants in the activity using floor control actions, although limits to the use of all the features can be established using floor control policies. Employees usually have little or no previous knowledge of training software, so the user interface must be as intuitive as possible.

The remainder of this paper is organized as follows. The specific requirements that a synchronous e-training platform must fulfill are enumerated in Section 2. In Section 3, related work is discussed thoroughly. The architecture of the proposed synchronous e-training platform is detailed in Section 4. Scalability tests and a case study of the use of the platform are commented in Section 5. Finally, Section 6 contains the concluding remarks and outlines future work.

2 Platform requirements

A synchronous e-training platform must fulfill various requirements. In addition to the pedagogical requirements that a common synchronous e-learning platform must satisfy, the particular environment in which the platform is used imposes some extra requirements. These requirements play an important role in the platform design.

The change from traditional on-campus training to distant training based on computer tools must also be accompanied by methodological changes [1]. There must be a transition from instructor-centered training activities to learner-centered training activities. In this situation, learners become responsible for their own learning, while the instructor acts as a facilitator, so learners participate actively in the learning process; no longer relegated to the role of passive spectators in expository lectures [30].

The constructionist learning theory suggests that learners construct meaning from existing mental structures [40]. The instructor must facilitate knowledge construction by promoting collaboration between learners in order to engage them in the training activity, so meaningful learning is possible. Thus, the use of a synchronous e-training tool fosters learning communities between learners and instructors. Collaborative learning has proven to be more efficient in improving critical thinking and communications skills of learners than the traditional lecture-based learning methodology [36].

The exhaustive analysis of commercial tools carried out in [20] concludes with the most common features that a synchronous e-learning tool must provide. However, e-training activities can be successfully developed using a subset of these features. In fact, a deliberate reduction of functionality depending on user skills and the environment in which the tool is used may be suitable.

One of the most important challenges that the human resources department must address is the possibility of employees' initial rejection of information and communication technologies (ICTs), in particular on the part of elder employees who have little or no knowledge of ICTs. These employees are usually against changes in the traditional instructional methodologies of training activities. Therefore, an e-training tool must be simple and intuitive, to avoid the cognitive overload of low skilled users. Nevertheless, the functionality provided by the tool must allow collaborative work between learners during e-training activities.

Many synchronous tools (e-training, e-learning, e-meeting...) require a producer or moderator in order to manage interactions between users. It is very common that few users interact at the same time due to network or training policy restrictions. The producer is responsible for developing the floor control of the session. A floor is a temporary permission to access or manipulate a specific shared resource. Thus, the producer grants or revokes floors to different participants in synchronous activities, so the relation between the producer and the instructor in an e-training activity is very important [34]. In fact, synchronous e-learning tools do not consider a producer, as they delegate the producer's responsibilities to the instructor. This reduces the cost of synchronous e-training as there is no need for specialized staff. However, this implies an extra workload for the instructor, who must guide both the activity and the floor control within it. For this reason, floor control must be simple so the instructor can focus his attention on the learning activity.

In addition to floor control management, there are also other aspects affecting the instructor's workload:

- Number of learners: the higher the number of learners in a training activity, the more difficult it is for the instructor to deal with learners' requirements and questions.
- Characteristics of learners: the personality of learners influence their participation level during the session. Extroverted learners participate more actively than introverts.
- Learning strategy: a guided learning strategy requires more attention from the instructor than autonomous learning strategies.
- Training subject matter: if learners have little knowledge about the material being taught, the number of questions will be higher, and the session will be regularly interrupted to solve learners' doubts. The difficulty of understanding the subject matter also affects the number of questions from the learners to the instructor.
- Feature used to get feedback: a learner can raise questions to the instructor in multiple ways. For example, a question can be posed by typing an instant message or it can be posed orally. The latter implies an interruption of the activity, while the former does not disrupt the flow of the class as the instructor may address the question at his convenience. Furthermore, several questions may be posed simultaneously using instant messages, while floor control operations are needed if questions are posed orally, which increases the workload of the instructor.

Ocassionally, e-training activities are carried out using different specific tools. A videoconference tool might be used along with a shared whiteboard and an instant messaging tool. Although the use of multiple tools adds flexibility to the e-training session, it also entails additional efforts for low skilled users. An integrated environment providing all the functionality needed during the activity is more appropriate, as users do not need to deal with the application windows layout in the desktop.

The restrictions imposed by the corporate network must also be considered. Large multinational companies are usually spread across several countries, so their networks constitute Wide Area Networks (WANs) which are organized in multiple Local Area Networks (LANs) interconnected by the Internet as shown in Fig. 1.

In most situations the available bandwidth in the links that connect the LANs of the various sites of the corporate network is fitted to the daily necessities of the corporation, without considering the additional load generated by synchronous e-training activities. Therefore, any multimedia application operating with this kind of network should use the available bandwidth as efficiently as possible. Multimedia traffic must not interfere with the overall operation of the corporation.

Moreover, industrial corporations usually deploy specific networks to support their manufacturing processes. The traffic flowing in these networks is critical and must be prioritized over other kinds of traffic. When this high priority traffic flows through the general-purpose network of the corporation it results in a narrowing of the available bandwidth for synchronous e-training activities.

It is also essential that the maintainability of the e-training platform be kept to a minimum to control the cost of training activities. This may eliminate the need for media servers to distribute multimedia data among the participants in an e-training session, which also implies that there is no need for specialized staff to manage them.

2.1 Features

According to these pedagogical principles and requirements, an e-training platform must provide the following features:

– *Audioconference*. The platform must forward the instructor's speech to learners. Audio quality must be as high as possible, as explanations from the instructor

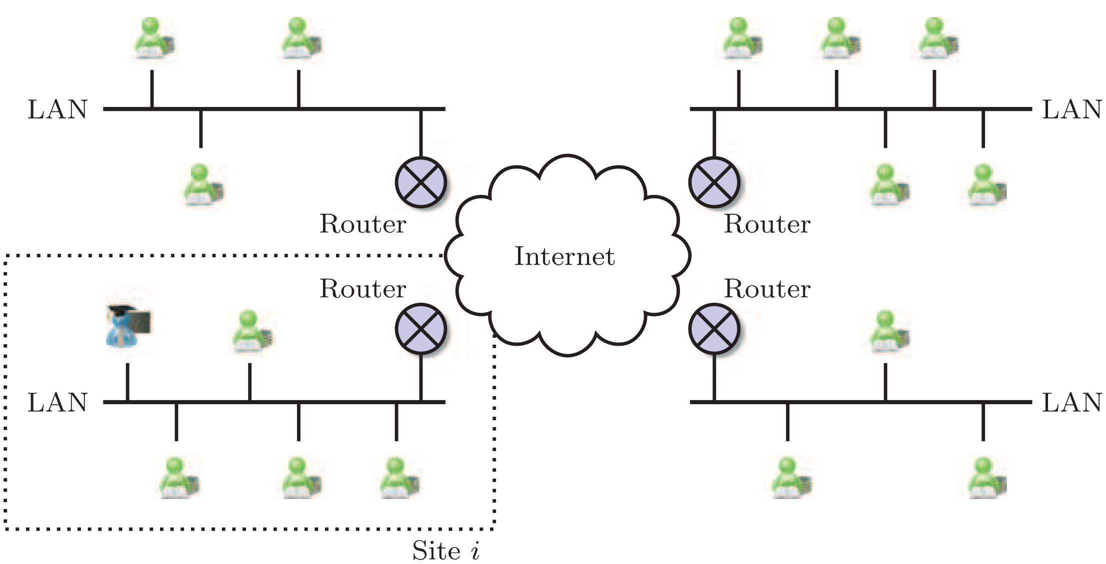


Fig. 1 Corporate network

must be clearly comprehensible. Thus, audio data must also be prioritized over other kinds of data, so difficult network conditions do not affect audio comprehension.

Similarly, the platform may offer learners an audio channel to participate orally in the e-training session. Multipoint audioconferencing enables participants in the session to collaborate in real time. A floor control mechanism is needed in order to avoid overlapping of audio streams from multiple participants.

The main drawback of multipoint audioconferencing is its high network bandwidth consumption. As a result, the number of simultaneous active audio channels must be a trade-off among network bandwidth consumption, intelligibility and interactivity.

– *Videoconference*. The video stream from the instructor usually portrays the instructor's talking head. As such, it does not provide any pedagogical content, yet the video stream reinforces the sensation of the presence of the instructor, and avoids a sense of isolation in the learners [11], [47].

  In addition, multipoint videoconferencing can be used to emulate face-to-face interaction between participants in the e-training session. In fact, non-verbal communication is possible, although video streams must have an appropriate temporal resolution. In this way, the instructor can see when a learner's facial expression reflects doubt.

  However, the high bandwidth requirements of video streams, even higher than audio streams, limits the number of simultaneous active video streams. When the underlying network has low-bandwidth links, only one participant, normally the instructor, can use the video channel. If more than one participant can use the video channel, it is necessary to share it using floor control operations.

  The video stream from the instructor can also be used to illustrate the steps of a manual process or to show last-minute information, such as printed documents, pictures, mechanical pieces, etc. In this situation, both the temporal and spatial resolution of the video stream must be high, so the network bandwidth requirements will also be high.
– *Instant messaging and presence control*. Presence control is absolutely essential in order to track attendees to an e-training activity. Instant messaging is very useful to provide feedback from learners to the instructor and to enable interactions between learners with minimum network bandwidth consumption. Instant messaging can even be used with severe network conditions.
– *Shared whiteboard*. The shared whiteboard is one of the most important features, as it is used to share the training material between learners and instructor. Most of the shared whiteboards used in synchronous e-learning offer common functionalities:

  – *Annotations*. The instructor can annotate the slides in the shared whiteboard to provide extra explanations. Annotations are also useful to communicate last-minute information. Learners can also annotate the shared whiteboard to pose questions to the instructor or to other learners. Handwritten annotations enable collaborative work between participants in e-training activities, as they can be used to jointly draw diagrams or sketches. In this case, floor control operations are needed to serialize the use of annotations.
  – *Telepointers*. A telepointer allows the instructor to point to a specific region in the shared whiteboard. This is useful to emphasize a concept in the current

slide. Learners can also use telepointers to pose questions to the rest of the participants.

- *Shared desktop and applications*. These features are very useful for an instructor to show how an application is used. By sharing the application, learners can observe how the instructor interacts with it. Sharing an application allows the instructor to show contents not directly available through the shared whiteboard. If a document format is not supported by the shared whiteboard, the instructor can share the document with learners by sharing its authoring tool. The main drawback of this solution is the high network bandwidth consumption, as these functionalities are usually implemented using video streaming.
- *Synchronous navigation*. This feature allows the instructor to guide the web navigation of learners. The instructor specifies which web pages the learners should visit, so all learners are viewing the same web page as the instructor simultaneously. This requires very low bandwidth, as only navigation commands, such as URLs, are sent from the instructor to learners. The instructor can also accomplish synchronous navigation by sharing the browser application with learners, but this solution is more bandwidth-consuming, since video streaming technology is used to deliver the contents of the instructor's browser window to learners.
- *File transfer*. The instructor may use the file transfer feature to deliver last-minute contents to learners. This is useful to distribute contents that are not shareable through the shared whiteboard.
- *Emoticons*. Any participant in a synchronous e-training activity may keep the rest of the class informed about his emotional state using emoticons. Some emoticons can be used to express agreement or disagreement.

Table 1 summarizes the features that can be found in a synchronous e-training platform. They are classified in three categories according to their relevance in developing training activities.

2.2 Interactivity

The aim of a synchronous e-training platform is to translate the face-to-face interactions of a traditional on-campus class to a virtual environment. They may be emulated using different media such as audio, video, shared whiteboards, and instant messaging.

A synchronous tool transmits multimedia data between participants in real time. The instructor's explanations are delivered to learners, and then feedback from learners is delivered to the instructor in order for the instructor to adapt the pace

Table 1 Features for synchronous e-training

Essential	Optional	Unnecessary
Audioconference	Videoconference	Shared desktop
Presence control	Telepointers	Synchronous navigation
Instant messaging	Shared applications	File transfer
Shared whiteboard		Emoticons
Annotations		

of the class or solve any doubts. Data can also be shared between learners in order to make collaborative learning strategies possible.

Ideally, all these kinds of interactions between participants in an e-training activity are possible. The synchronous tool used to support the training process would allow the use of all features in a multipoint format. However, restrictions, especially those related to the underlying network and the need to serialize the use of shared resources, make multipoint use of all features difficult or impossible.

Firstly, the available bandwidth in the network links between participants may be very different and may vary over time. In addition, the e-training platform must provide as many forms of interaction as possible but in an ordered manner, so collaboration between participants is easy and worthwhile. Nevertheless, the network resources used to transport data are directly related to the interactivity level of the e-training activity. The higher the interactivity level, the more network resources are used. Thus, the interaction mode of the features of the e-training platform is a trade-off between interactivity and the resources used.

Three interaction modes can be proposed for each feature of the e-training platform:

- *1-N mode*. There is a single data source and N receivers under this mode. Webcasting is a representative example of this form of interaction: data flows one-way from the sender to the receivers. The instructor plays the role of the sender, while learners are the receivers. Learners are not able to provide feedback to the instructor. Figure 2a illustrates this interaction mode.
  The main advantage of the 1-N interaction mode is the low amount of resources used. If IP multicast is available, data replication is not needed. Furthermore, there is no need to define floor control policies, as only the instructor can send data. However, the lack of feedback from learners makes this mode only suitable for one-way features.
- *N-1 mode*. In this mode data flows in the opposite direction: from N sources to a single receiver. Learners send their data streams to the instructor, who is the single receiver of all of them. In this way, learners are able to provide feedback in the form of questions and doubts to the instructor. However, the role of the instructor as a facilitator of the learning process is eliminated, as he cannot communicate with learners. Figure 2b represents the flow of communications under this interaction mode.

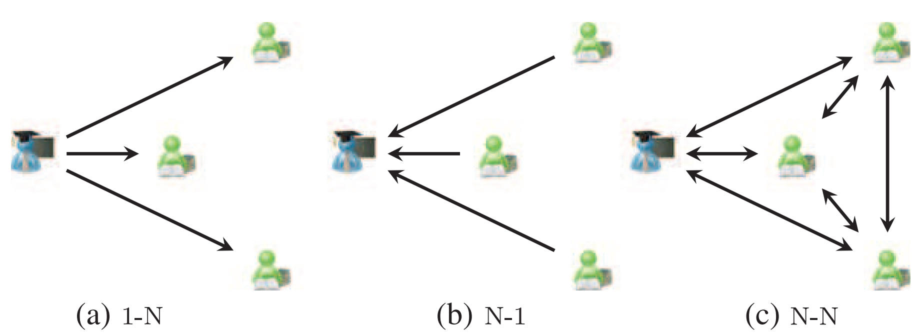


Fig. 2 Interaction modes

– *N-N mode*. Using this interaction mode all the participants in an e-training activity can interact with each other, so collaboration is possible. Not only can learners communicate with the instructor, but they can also communicate amongst each other. Figure 2c illustrates all the interactions possible under an N-N interaction mode.

An N-N interaction mode implies N data sources and N receivers. All participants send and receive traffic. If IP multicast is available, there are N data streams flowing from senders to receivers with no replication. However, when IP multicast is not available, unicast communications must be used and the number of data streams that must be sent by senders increases exponentially as shown in (1):

$$N \times (N - 1) = N^2 - N \tag{1}$$

The N-N interaction mode can be relaxed by using floor control. A floor may be used for each feature, so interactivity is restricted by the maximum number of participants that can hold each floor simultaneously. For example, if floor control policies specify that a maximum of two simultaneous participants can use the audio channel, there will be either zero, one or two audio streams flowing to participants at any given time. This represents a 2-N interaction mode, which allows the instructor and one learner to participate orally in the class simultaneously.

However, floor control adds extra load for the instructor, as there is no moderator in charge of floor control operations. Thus, floor control must be as simple as possible. The selection of the interaction mode for each feature is based on the need of the feature to provide feedback. There must be a trade-off between interactivity, network resources used and workload for the instructor due to the floor control management. For instance, a high interactivity level is achieved by the instructor granting the use of the audio channel to many learners, but this requires a significant network bandwidth to transmit all the audio streams to the rest of participants, and also implies a high workload for the instructor, who must grant and revoke floors. In contrast, if the instructor is restricted to granting the audio floor to few learners, the interactivity level is low but less network bandwidth is used and the workload of the instructor, granting and revoking floors, decreases.

3 Related work

Before addressing the development of a new platform for synchronous e-training, the solutions proposed in the technical literature are analyzed.

The multicast backbone (MBONE) [10] has promoted the development of multiple conferencing and collaboration tools, such as vat [29], vic [37] and wb [17]. These tools are used for audioconferencing, videoconferencing and shared whiteboard respectively. While there have been successful experiences using these tools [16, 18, 35], the simultaneous use of many different tools made participation in these synchronous sessions difficult for low skilled users.

A pioneer tool was developed by Latchman et al. [31]. This tool is designed to capture the audio and video of the instructor while he is teaching a traditional class and transmit them to remote learners. As the resolution of the video is low, this system uses two additional cameras to capture the class notes and the instructor's handwritten notes. There is also a chat window in the interface of this tool that provides instantaneous communication between all the participants in a class. This design, based exclusively on video streams, is now obsolete. Today, class notes are always electronic documents (ppt, pdf, word, gif…) to avoid problems with video resolution, while handwritten notes may be made directly on the class notes to enhance the learning process. In [32], the authors present several improvements and the use of the tool in blended activities, combining synchronous and asynchronous e-learning.

Isenhour et al. [26] proposed Virtual School, a hybrid synchronous and asynchronous open tool for collaboration of learners in the context of the classroom environment. It provides a persistent chat that saves messages for review of previous discussions. The videoconferencing facility is supported by third-party conferencing tools. Notebooks are the main component of Virtual School, as they allow collaborative authoring of class assignments between learners. Notebooks support shared whiteboards and collaborative editing of HTML pages and lists of external references. However, this tool is limited to an on-campus environment, so learners cannot communicate from their homes.

Deshpande et al. [13] developed a real-time interactive virtual classroom. This system manages the electronic slides in a particularized manner using a client/server subsystem called SlideCast. The virtual classroom also integrates a sophisticated coding to handle the video of the instructor when it contains handwritten information. In this system, the interface of learners consists of a specific client to receive the video of the instructor and his handwritten notes and a separate web browser with a plug-in to receive the electronic slides. The use of separate windows is an undesirable characteristic.

Bouras et al. [8] developed a new full-integrated system that provides audio, video and chat features. It includes an original control module, which can operate on ISDN and IP networks. However, documents are shared between users using video streaming, which requires high network resources if the number of users increases.

Fung and Ledesma [19] reported on the utilization of the VITLE platform (Virtual Integrated Teaching and Learning Environment) in Hong Kong during the respiratory syndrome outbreak in April 2003. This tool provides a simple and effective user interface, although the performance of the whiteboard is limited. It is totally based on Adobe Flash® technology requiring intermediate Flash Communication Servers.

Snow et al. developed the Network EducationWare (NEW) system [45] to provide synchronous distance education via the Internet. This system is easy to set up and operate, open source, totally free for educational institutions, and capable of operating acceptably with a reduced bandwidth. It has been constructed using pre-available tools, including MBONE conferencing tools, by adding an interface window for each tool. These tools communicate with a Transport Layer Multicaster to send and receive information to and from other clients. A floor control module is also required to coordinate the operation of the system. The use of floating windows makes the interface difficult to use for low skilled users.

Zhao et al. proposed a highly interactive system for distance learning [50] that emulates traditional classrooms. This tool is totally focused on collaborative learning and includes several subsystems. The video subsystem is used to deliver the image of the instructor, slides and educational videos to learners. The audio subsystem is used to transport the voice of the instructor to learners and some feedback from learners to the instructor. The seminar subsystem is used as a discussion area, in which the learners and the instructor can interchange videos, audio, text messages and any other educational content. All the subsystems are managed with a control panel. The control panel and each subsystem require one window on the computer screen, but the seminar subsystem can use even more windows. All these independent windows must be arranged on the screen by learners. Zhao and Zhang proposed the virtual classroom model [49] which is integrated in their distance learning system.

Higuchi et al. [24] developed an interactive multimedia instruction system called IMPRESSION. Educational multimedia materials are stored in several public web servers; the instructor can download them to his terminal and modify them. All the operations on the materials are sent to attendee terminals through a lecture server, so attendee terminals are synchronized with the operator terminal. However, this system is only designed to share multimedia material through the Internet, and an independent videoconferencing application is necessary to transmit video and audio between the participants of a class.

Yang and Liu developed the WVOC (Web-based Virtual Online Classroom) framework [48]. It is composed of an Instructional Communicating Environment (ICE) and a Collaborative Learning Environment (CLE). The ICE supports the synchronous interactions between instructors and learners using video broadcasting, chat and electronic whiteboards, while the CLE provides tools for active learning. This framework uses video streaming exhaustively to deliver lectures to learners and requires a complex infrastructure composed of a broadcaster center and servers.

El Saddik et al. developed PECOLE [15], a peer-to-peer tool for multimedia collaboration. It provides functionality such as multipoint audio and video, synchronous navigation and chat. However, floor control policies are not applied to audio and video, so any learner may activate his audio and video anytime. Therefore, this tool is not suitable for low bandwidth networks.

Another interesting tool is ConferenceXP [39]. This is a non-commercial platform oriented to collaborative and distance learning. It provides an Application Programming Interface for the development of custom functionality. It is especially focused on supporting collaborative activities within a classroom using tablet PCs, but it can also operate in a remote manner using multicast in Internet2 [2].

In [33] the authors proposed a virtual classroom based on a client/server model with a video streaming server and many clients using Adobe Flash® technology. They combined synchronous and asynchronous e-learning features. The main drawback is that a user must master various applications to participate in learning activities.

Premchaiswadi developed a blended platform combining synchronous and asynchronous e-learning [42]. The platform is composed of a virtual interactive classroom and a separate collaborative workspace. The virtual classroom supports the synchronous interactions between instructor and learners using audio and video streaming, content presentation, whiteboard, instant messaging and presence control. However, floor control policies are not applied to the whiteboard, so any learner may draw or erase any item at any time. Furthermore, the scalability of the platform might be compromised as all the synchronous features are managed by a single streaming server.

Another e-learning platform based on video streaming was presented by Ullrich et al. [46]. Real-time lectures are streamed to learners' mobile phones, including the audio and video from the instructor, didactic content, and annotations made by the instructor on the content. The instructor receives feedback and questions from learners in the form of short messages. The system also includes a remote monitoring mechanism to control learners' activities.

Bijlani et al. developed A-VIEW [4], a videoconferencing and telepresence system oriented to e-learning, providing features such as video and audioconference, instant messaging, document sharing, whiteboard and annotations. Several of these features can run in separate display windows. This enhances the learning experience in large classrooms, since a projector can be used to display each feature separately to the audience. The main drawback of A-VIEW is that it is limited to a maximum of two users interacting simultaneously.

None of the aforementioned tools is specifically oriented to e-training or profits from the characteristics of corporate networks. These characteristics may be taken into account to implement efficient delivery techniques. In the same manner, user profiles must be considered when designing the functionality and interface of the e-training tool. Furthermore, the e-training platform must be connected with the asynchronous e-training portal of the corporation. A global e-training solution must combine synchronous and asynchronous tools. The limitations observed in the described tools have motivated the development of the e-training platform detailed in the following sections.

4 Platform architecture

The e-pSyLon (**e**lectronic **p**latform for **Sy**nchronous **L**earning and Collaborati**on**) e-training platform is composed of various entities: a client tool, a floor control server, a Rendezvous Point (RP) acting as a Session Initiation Protocol (SIP) focus and zero or multiple optional Real-time Transport Protocol (RTP) relays. The client tool allows learners and instructors to participate in e-training activities. The floor control server implements the floor control policies. The RP authorizes users to participate in e-training activities. It also negotiates the configuration of the multimedia sessions to be established during the activities. Finally, multiple RTP relays may be used at the transport level to interconnect several native multicast islands. E-learning activities can be carried out using client tools exclusively without deploying any RTP relay, floor control server or an RP, providing that IP multicast is available in the underlying network and the virtual class is self-moderated. In this case, the client tools of the participants of the e-training activity can communicate peer to peer.

Figure 3 illustrates the organization of the e-training platform. The e-training platform is organized into four virtual networks: the relay mesh, the signaling network, the mesh control network, and the floor control network. The relay mesh connects all the RTP relays and the clients in the e-training activity forming an overlay network to deliver multimedia content efficiently (thick dashed lines). The signaling network connects the RP with the clients using the SIP protocol [43] (thin dotted lines). The SIP protocol is used to establish and tear down the multimedia RTP sessions associated with an e-training activity. The mesh control network is established between the RTP relays and the RP (thick solid lines). The relay mesh can be reorganized to provide efficient delivery of data. Finally, the floor control network interconnects the clients, both the trainer and the trainees, with the floor

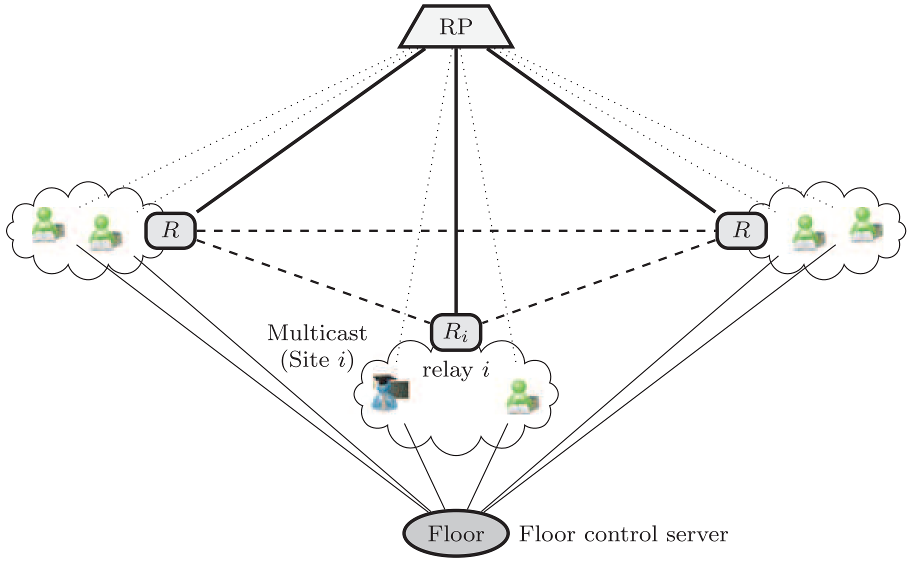


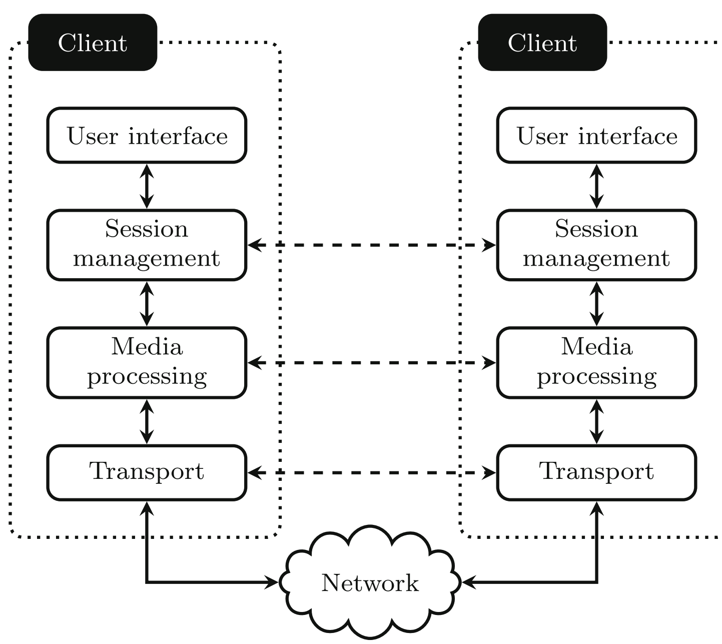
Fig. 3 Architecture of the e-training platform

control server (thin solid lines). This architecture is described in depth in [38]. This work focuses on the design and operation of the client tool.

In contrast to other similar solutions, the client tool is a complete application in itself rather than an aggregation of independent tools. This results in an easy-to-use integrated user interface. Furthermore, it is possible to optimize those parts considered vital for the correct operation of the application, as all data processing is under control.

The design of the client tool is organized in layers, as seen in Fig. 4. For simplicity the same tool is used by instructors and employees, with different privileges depend-

Fig. 4 Layered architecture of the e-training client



ing on the role of the participant in the e-training activity. In the following sections the details of each of these layers are explained.

4.1 Data transport

The aim of the transport layer is to provide end-to-end delivery of multimedia data between participants in e-training activities. This is accomplished using the Real-time Transport Protocol (RTP) [44]. Data from each feature (audio, video, annotations on the shared whiteboard, etc.) of the e-training activity is carried in a separate RTP session.

Although RTP is designed for the delivery of continuous data streams such as audio and video, it can also be used to deliver non-continuous data streams such as text messages and annotations. This has a series of advantages. Firstly, RTP can be used over UDP and IP multicast, so multicasting techniques can be employed to deliver data, saving network bandwidth. Secondly, RTP timestamps can be used to synchronize all features of the e-training activity. Typically, audio and video must be synchronized accurately, but all the other features can also be synchronized, which results in a more successful user experience. Furthermore, e-training activities can be recorded and later replayed with temporal adequacy. Thirdly, RTP characteristics, such as sequence numbers, can also be used with non-continuous media. This allows for the detection of packet loss, packet duplication and packet reordering. Similarly, as all RTP data stream are labeled with an identifier, any data stream can be associated with its owner participant. Finally, some routers in the network are aware of RTP traffic, so it can be prioritized over other kinds of traffic. Thus, a different quality-of-service can be applied to data depending on its nature [5].

Typically, the RTP session used to convey data from a specific feature contains one or many data streams. When the feature is used with a 1-N interaction mode, there is one source that can send one or many data streams to the RTP session, while N data sources exist when using N-1 and N-N interaction modes.

Ideally, data from each feature is carried in a separate RTP session associated with an IP multicast group. Thus, participants in an e-training activity join these RTP sessions by joining their multicast groups to send and receive multimedia data. However, IP multicast is rarely available throughout corporate networks. In this case, one or multiple relays can be used to forward RTP traffic between IP multicast and non-multicast network domains. An example of the use of four relays to communicate participants in an RTP session is shown in Fig. 5. In [22] an efficient technique for the delivery of multimedia data in corporate e-training activities based on RTP relays is proposed. The relays are always deployed in a full-mesh topology, so latency is minimum, as two participants are separated by two relay hops maximum.

Continuous media are more resilient to network issues than non-continuous. Typically, audio and video codecs are able to cope with packet loss. In contrast, a lost packet in non-continuous media, such as instant messaging or the shared whiteboard, is unacceptable, as it may carry a text message with a question from a learner or a control operation in the shared whiteboard. For this reason, Forwarding Error Correction techniques are used with non-continuous media to recover from packet loss.

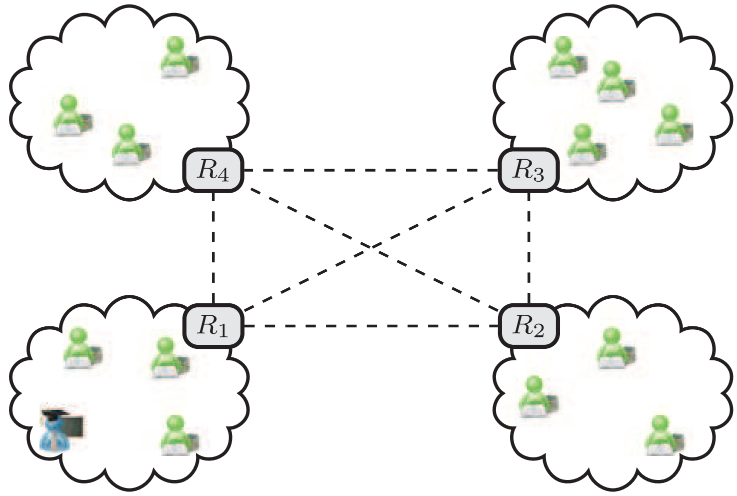


Fig. 5 Deployment of RTP relays in a full-mesh topology to extend the scope of an RTP session

4.2 Media processing

The media processing layer is responsible for the management of all media streams in an e-training activity. This comprises the establishment of the RTP sessions and the management of all data structures and operations related with the features provided by the client tool.

Moreover, this layer creates and consumes data streams conveyed in the RTP sessions. This implies a conversion between network data and logical structures and operations. Once multimedia data streams are processed they can be translated to user interface events.

4.2.1 Audio and video

The transmission of continuous media through IP networks faces many challenges, as the network only provides a best effort service [41]. Packet loss, packet reordering and packet duplication must be dealt with by a multimedia networking application. However, the most important challenge for multimedia networking is interarrival jitter due to the varying transit time of packets.

Audio quality is highly affected by network issues. Jitter and packet loss produce audio chopping which makes interactive audio conversations extremely difficult to maintain [6]. Video quality also decreases when packet loss and jitter appear, resulting in video artifacts. Nevertheless, in contrast to audio streams, video artifacts do not prevent the rendering of video streams. Several case studies have indicated the importance of audio quality over video quality in collaborative environments [25]. The client tool uses a reception buffer to compensate for jitter. The buffering time varies from 40–80 ms. Furthermore, a simple lip-synchronization algorithm is used to maintain synchronism between audio and video. Since video is usually transmitted at a maximum rate of 15 frames per second (it can be configured in the client tool), the accuracy of the synchronism need not be high, so the algorithm drops or copies video frames when the audio and video get out of sync. Audio streams are never delayed, so real-time interactions are not disturbed.

Each participant in an e-training activity can use the audio and video channels if the floor control policies permit. The instructor may grant or revoke the audio and video floors to any participant. There is a maximum number of participants who can use the audio and video channels simultaneously to avoid excessive network bandwidth usage. This number also depends on the intelligibility of data streams. If multiple participants transmit audio at the same time, the overlapping of the audio streams will generate unintelligible information. Although multiple video streams do not overlap like audio streams, only a few can be rendered on screen.

An inter-destination synchronization is indirectly achieved by the instructor granting floors to learners. For example, the instructor can ask a question to learners using the audio channel and subsequently grant the audio floor to those learners that wish to answer the question. Although there can be slightly differences in the playout points of the audio channel in the learners, these are negligible compared to the response time of the instructor to grant floors. Thus, the instructor can guarantee the fairness, as all of them have the possibility to answer the question.

Usually, a webcam and a microphone are employed as the source of video and audio data streams respectively. These streams are encoded before they are passed to the transport layer. The codecs used must have specific characteristics. The codecs must provide a high compression ratio to use the minimum network bandwidth possible. Similarly, the quality of decoded signal should be high, so signal degradation is minimum. The codecs must encode and decode with low latency in order to operate in real time. Furthermore, they must be packet loss resilient to cope with packet loss.

The client tool can use several audio codecs to encode and decode audio data. These codecs are especially oriented to encode human voice sampled at 8000 Hz. At its current stage, the tool may encode audio using G.711 a-law, Speex (8 modes) and iLBC (2 modes). Audio streams are usually encoded with iLBC codec with a packet size of 20 ms, resulting in a bitrate of 15.2 kbps. The overload generated by transport protocols (RTP/UDP/IP/Ethernet) results in a network bandwidth consumption of 38.4 kbps for each audio stream.

Video streams are encoded with the VC-1 codec, which is a video codec used to encode video from a very low bitrate to high definition video. The simple profile of VC-1 is used. The video framerate can vary from 5 to 15 frames per second and the resolution can vary from 160×120 to 320×240, which yields a bitrate of 50–300 kbps for each video stream, and 52.8–308.4 kbps in the network.

4.2.2 Instant messaging and presence control

Instant messaging is implemented according to ITU-T T.140 [28] recommendation and RFC 4103 [23]. Text messages are shared between all the participants in an e-training activity through a specific RTP session. Furthermore, the RTP specification defines a complementary control protocol called Real-time Transport Control Protocol (RTCP). The operation of RTCP is based on the periodic delivery of RTCP packets between all users in an RTP session. A simple presence control is implemented using RTCP rules for maintaining membership to the RTP session.

4.2.3 Shared whiteboard

The whiteboard feature is designed according to the ITU-T T.126 [27] recommendation. It is composed of many workspaces, one for each document being shared. A workspace consists of a number of stacked planes with the same number of pages. The background plane is the slide presentation plane, which contains the shared document. One or multiple annotation planes are located in front of the background plane. Finally, multiple telepointer planes may be located in front of the annotation planes. This is illustrated in Fig. 6.

Every whiteboard workspace has a slide presentation plane. This plane represents the most important educational information of the workspace. The contents of this plane may be a blank page to emulate a whiteboard, or a PowerPoint, PDF or SVG document can be used as the source of the slides. In this case, it is necessary that all participants have the presentation downloaded before the beginning of the e-training activity. Thus, there is no need to synchronously deliver the whole document to all the participants in the activity; only operations to synchronously navigate through the pages of the workspace must be delivered. This is highly appropriate to low bandwidth environments. This is not a restrictive requirement, as corporations usually have a web site where educational documents can be uploaded so learners can download them before the start of a scheduled e-training activity.

An annotation plane contains the data structures required to manage annotations for a single user. Freehand strokes are the most common and useful type of annotations in this kind of applications as both text and drawings are possible. It is very useful to associate an annotation plane to every participant in an e-training activity to differentiate annotations from different users, so there are as many annotation planes as participants in the activity within a single workspace ordered using a Z-index. In this way, planes may be hidden by the instructor using floor control operations to disable annotations from participants with a computational complexity of O(1).

A telepointer plane represents a cursor which can be used to point to any location within a whiteboard workspace. It provides the icon, position and size of the cursor; as well as the state of the cursor buttons. To implement this feature the RFC 2862 [12] is followed. As well as the annotation planes, every participant in an e-training activity has one telepointer plane for each workspace. The telepointer planes are also sorted in the Z-index.

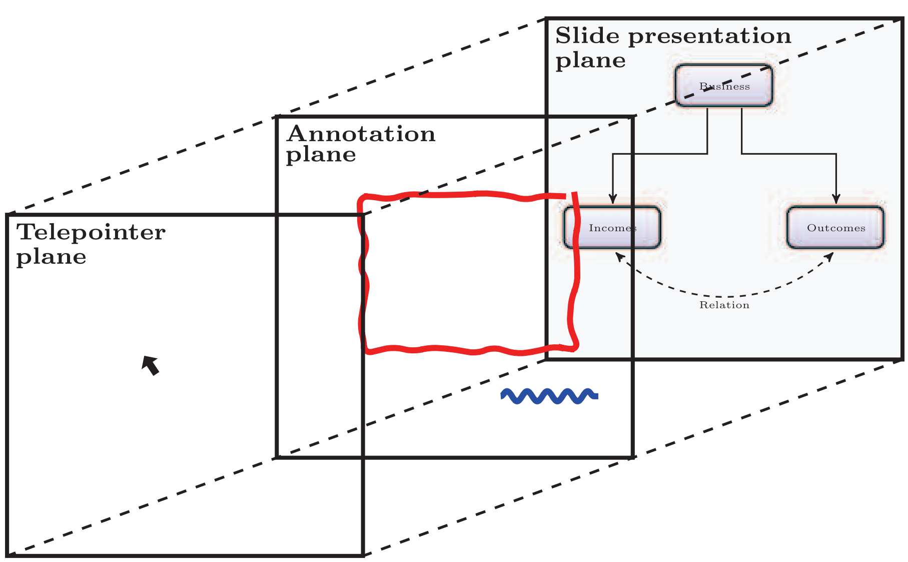


Fig. 6 Organization of a shared whiteboard workspace in planes

Although the workspace is composed of multiple pages, not all the information of the workspace can be seen at the same time. Thus, a common view of the workspace is shared between all the participants to synchronously navigate through the contents of the workspace. The view is defined as the rectangular region of the whole workspace that is being visualized by all the participants including the elements belonging to the non-hidden planes that are placed within the region. The most important attributes of the view are the current page, the zoom level and the displacement between the top-left corner of the workspace and the top-left corner of the view as illustrated in Fig. 7. A method for proper visualization of synchronous workspaces is detailed in [21].

The creator of a whiteboard workspace (instructor or learner) controls the common view of the workspace shared by participants. In this way, he can change the page and the zoom level of the workspace for all the participants, so the view is synchronously updated. Furthermore, there is a floor that can be granted to a participant so he can freely navigate through the workspace. Nevertheless, whenever this floor is revoked, his workspace view is re-synchronized with the view of the workspace owner.

In order to transport data related to the planes which compose the workspaces, three different RTP sessions are used: the shared whiteboard session, the annotation session and the telepointer session. The translation from data network entities (RTP sessions and RTP streams) into data applications objects (workspaces and planes) is represented in Fig. 8.

The shared whiteboard RTP session conveys information about the active workspaces in the whiteboard and their background planes. Each RTP stream in this session maps a single workspace. As documents to be shared must be downloaded previously to the beginning of an e-learning activity, only workspace navigation events, such as page or zoom factor changes (changes in the common view), are carried in an RTP stream.

The annotation RTP streams represent annotation planes. Each annotation plane is associated with the workspace to which it belongs using a workspace identifier carried in the RTCP periodic packets of the stream. Annotations made by participants are carried in RTP data packets.

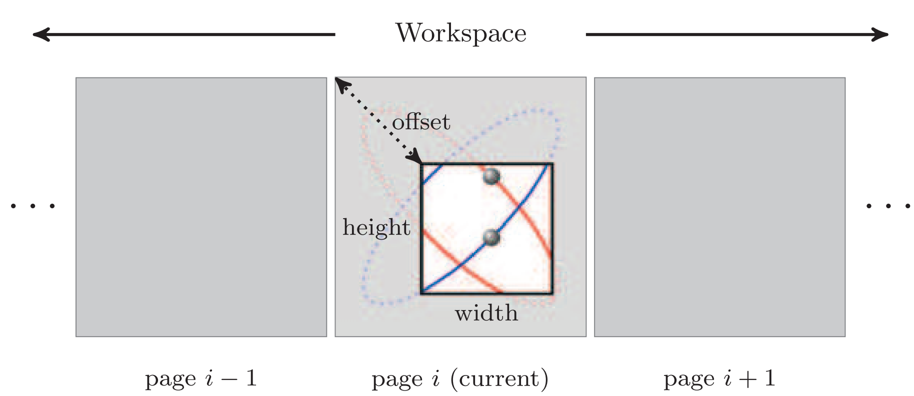


Fig. 7 View of the workspace

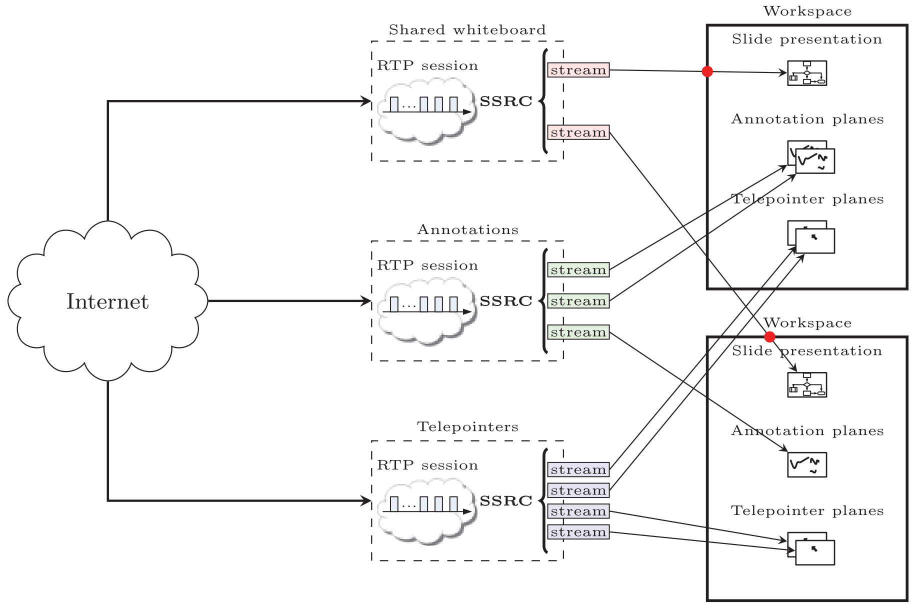


Fig. 8 Transport of the elements of the whiteboard over the network

Data streams within the telepointer RTP session convey information about telepointer planes. As with annotations, each stream represents a single telepointer plane and is associated with a workspace using an identifier carried in RTCP packets.

All groupware and collaborative tools must be able to accept late joiners [14]. Participants may join a synchronous activity once it has begun, so they need to be incorporated in the activity as quickly as possible. The whiteboard of the incoming participant must be synchronized with the rest. This is done with the periodic transmission of RTCP packets in the shared whiteboard RTP streams, which contain the URI of the documents being shared and information about the workspace views. Annotations made by participants previous to the arrival of a late joiner cannot be recovered, as it would be necessary to retransmit all annotations. This would be extremely complex when there are various participants joining at the same time.

Continuous media such as audio, video and telepointers do not need to deal with late-joining participants, as incoming data invalidates all previously received data.

4.3 Session management

This layer is responsible for establishing e-training activities. This is accomplished in three steps. Firstly, there must be a mechanism for discovering active e-training activities, so learners can join the desired one. Secondly, there must be a negotiation of multimedia configuration, such as the features to be used and the encoding of multimedia data. Finally, RTP sessions must be created in order to actually join the e-training activity and send and receive multimedia data to and from the rest of the participants.

The RP manages all the e-training activities in the platform. Each activity is described using an XML document as defined by the XCON centralized conferencing framework [3] with some extensions. This document enumerates the multimedia sessions of the activity, the users authorized to participate in the activity, the floor control policies, etc.

The adopted solution uses SIP to discover and negotiate available e-training activities and the Session Description Protocol (SDP) to describe e-training activities. Each activity is identified by a SIP URI such as `sip:act001@rp.arcelormittal.com`. Participants join an activity using the SIP standard handshake process, which makes the synchronous e-training platform inter-operable with most SIP applications and devices, such as voice over IP phones. Furthermore, it is possible to provide standard authentication mechanisms.

The multimedia configuration of an e-training activity is defined by an SDP description that enumerates the media used during the activity, transport information such as the multicast IP address (or the IP of an RTP relay) and the port of each RTP session, the encoding schemes being used, etc.

Learners usually enter the corporate website where there is a schedule of activities with a link to their SIP URIs. By clicking an activity link, the client tool launches and sends a SIP INVITE message to the RP to start the SIP handshake process. The RP is responsible for granting access and providing the SDP description for the activity. During the handshake process, the client tool obtains an SDP description of the activity which is used to establish the multimedia RTP sessions.

The SDP description of an e-training activity may be customized, so a specific multimedia configuration is applied tailored to the participant. In fact, the XML document used by the RP to describe an e-training activity specifies which features can be used by each participant. Thus, it is possible for a participant to join an e-training activity using a subset of all the features of those provided by the e-training platform. This is a flexible solution for heterogeneous participants. For example, a participant may have a very low bandwidth link, so he is not able to receive continuous media such as audio and video. Participants only join the RTP sessions described in the SDP description as detailed in Fig. 9. Instructor (A) and one learner (B) join all the RTP sessions, so they participate in the activity using all the available features. Learner C and learner D only use some of the features to communicate with the others. Neither of them can send or receive video streams to or from other participants, but they can communicate using the instant messaging feature.

The use of SIP for establishing and tearing down e-training activities has various advantages. Firstly, a standard procedure is defined for the negotiation of multimedia sessions among heterogeneous devices. This enables the participation of users in the e-training activities using a variety of devices such as the e-pSyLon client tool and voice over IP devices. Furthermore, dial-out scenarios where the RP initiates the communication with participants are possible, since SIP defines a mechanism for locating users. Finally, the integration of SIP with other Internet protocols such as HTTP facilitates the integration of the whole e-training platform within a Learning Management System (LMS).

E-training activities can easily be recorded using specific tools, as SDP is an international standard also used in audio and video streaming. In fact, a specific tool has been developed that takes advantage of RTP timestamps to record all media synchronously. In this way, activities may be recorded while they are taking place and saved as a reusable learning object based on the SCORM 2004 specification. Thus, recorded lessons can be directly imported to LMSs such as Moodle in order to provide a blended learning experience.

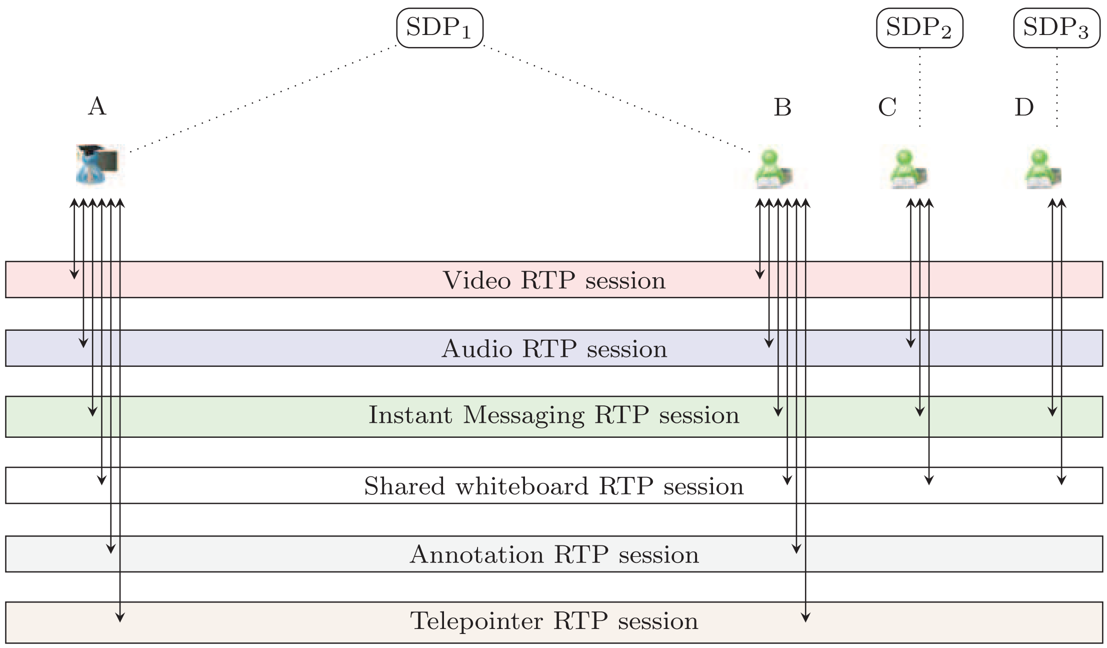


Fig. 9 Availability of features to participants depending on the SDP description

4.3.1 Floor control

Many features of the e-training platform are under floor control policies. Floor control is a means of managing joint or exclusive access to shared resources in an e-training activity. The Binary Floor Control Protocol (BFCP) [9] is used to implement the floor control.

BFCP defines a conference as a group of participants that communicate by accessing the media resources of the conference (audio, video and, etc.). Access to these resources is controlled by one or many floors. There are three entities in a BFCP conference that communicate to manage the state of the floors: participants, a chair and a floor control server. The chair requires the server to grant, deny or revoke floors, while the server satisfies these requests as long as the conference policies are not violated. Participants can request or release floors. The BFCP server manages all floor control operations, so it maintains a coherent state for all the floors. The protocol also supports functionality such as third-party floor requests and multiple chairs.

The e-pSyLon e-training platform includes a BFCP server to manage floor control interactions between learners and the instructor. Learners play the role of BFCP participants, while the instructor acts as a BFCP participant and chair at the same time. The floor control server grants, denies or revokes floors attending to the participants' requests and the instructor's indications.

The BFCP server obtains the virtual class floor control policies from the XML document of the activity before it begins. This XML document specifies the number of floors and their associated features, the chair of each floor (the instructor by default), the maximum number of granted participants for each floor, and which participants can make third-party floor requests (usually the instructor). Figure 10 shows the different entities involved in floor control.

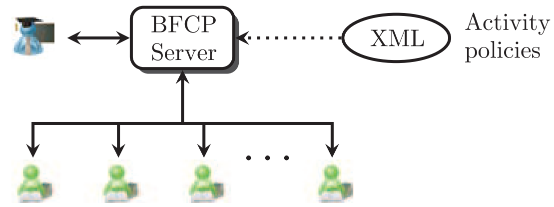


Fig. 10 Entities involved in floor control

In summary, the e-training platform defines several floors which allow instructors and learners to use the audio and video channels, send and receive instant messages, create whiteboard workspaces to share documents, make annotations on white-board workspaces, freely navigate through the contents of workspaces and use the telepointer.

4.4 User interface

The user interface has the same appearance at both the instructor and the learner sides. This results in a less steep learning curve when changing the user role, as a learner may become an instructor and vice versa using the same tool at any time. This situation would not be possible if a different tool were used.

The client tool is fully integrated with the web browser, so the tool launches when the users clicks on a SIP URI link in a web page. The SIP URI links of the activities are usually shown in the corporate LMS web site.

The first time the tool starts, a wizard appears to configure the multimedia devices of the computer. Then, the user must specify his SIP URI, name and password in order to contact the RP. Once the SIP dialog is established with the RP and the SDP description of the multimedia sessions obtained, the main window of the client tool appears.

The most commonly used commercial e-training tools have quite similar user interfaces. The design of the interface of the client tool follows the patterns observed in the most significant tools detailed in [20]. Figure 11 shows the implemented interface.

There are four areas between the menu bar at the top and the status bar at the bottom, each of them corresponding to a different feature. On the left side of the screen there are three small areas: the top area shows the video channel, where video streams from different participants in the e-training activities can be rendered; the central area shows text messages shared by the participants; and the bottom area shows the participants in the current e-training session and the floor control interface. The state of each floor for each participant is arranged in a matrix. The state of each floor for one participant is displayed in a row, while the state of one floor for each participant is displayed in a column. Using this interface, participants can request floors and the instructor can grant, deny or revoke them.

On the right side of the screen there is a large area for the presentation of whiteboard workspaces. This space is used to share documents, annotations and telepointers among participants. Each of the tabs in this area represents a single whiteboard workspace.

At the top of the presentation area there is a navigation bar, which allows participants in the e-training activity to navigate through the slides of the workspaces using several buttons. This bar also includes the *hand* button that can be used by

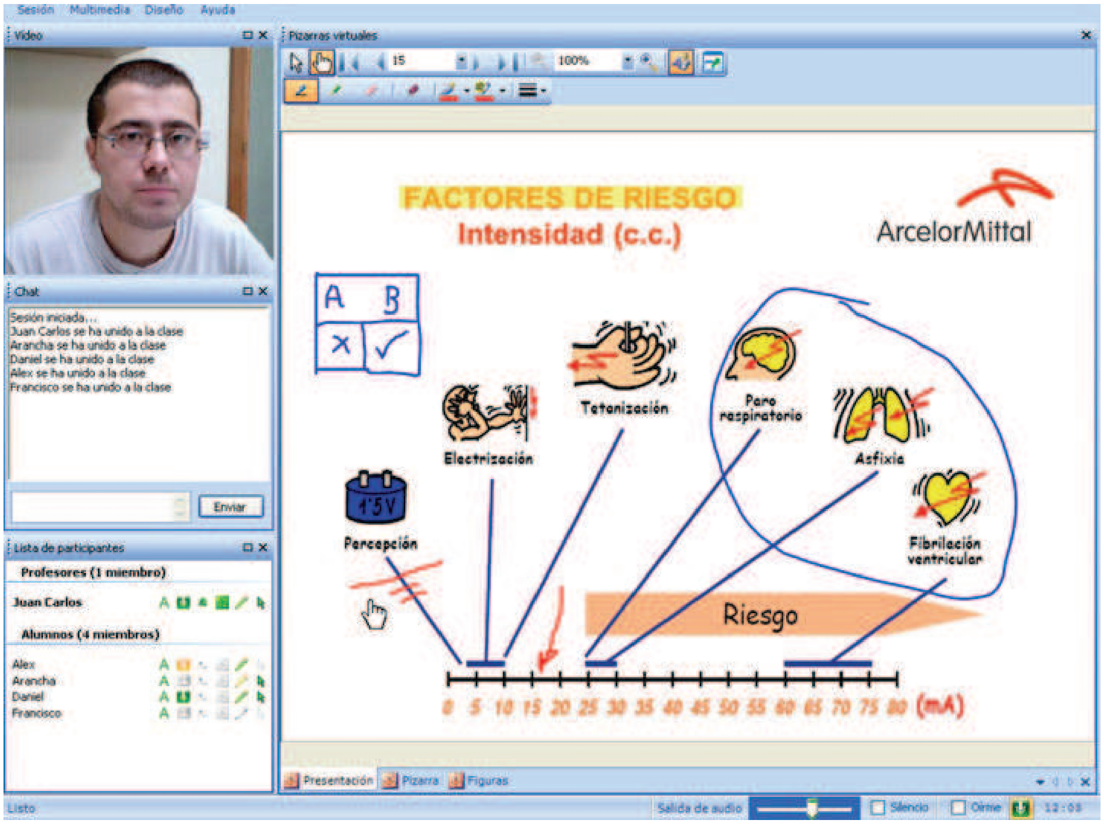


Fig. 11 A snapshot of the e-pSyLon client tool in use

any participant to request the associated floor to use a telepointer on the shared whiteboard.

Just below the navigation bar there is a drawing bar. Using the buttons on this bar, any participant can make simple drawings and annotations on the current slide to point out aspects of the information presented, provided that he is the holder of the floor related to annotations.

The client tool also offers various layout modes which may be selected by the instructor, so all participants in the activity share the same layout. In the full-screen mode the current whiteboard workspace occupies the whole screen. It is also possible to locate the column containing the video channel, the instant messaging and the presence and floor control windows on the right side of the interface.

4.5 Comparative with commercial platforms

The e-pSyLon platform has two advantages over commercial platforms. Firstly, it is designed taking into account the specific characteristics of corporate networks, usually organized into multicast islands, and the users involved in the training process. In contrast, most of the commercial platforms are general purpose synchronous tools that can be used for carrying out different synchronous events, such as online meetings or product demonstrations. In fact, many commercial tools are not specifically oriented to learning but to web conferencing. The e-pSyLon platform is able to reduce the bandwidth consumption by using IP multicast where available.

Thus, more interaction among users is possible, as a high number of multimedia streams can flow through the platform, providing an enhanced learning experience.

Secondly, the platform is able to adapt to low-bandwidth network environments. Its ability to provide different functionalities to users according to their requirements and capabilities allows users to participate in e-training activities regardless of their location or the characteristics of their network links. A user can also join an activity using a subset of all the available features of the activity in order to save bandwidth.

The main drawback of the e-pSyLon platform with respect to commercial platforms is that the latter usually offer additional functionalities such as shared desktop or file transference. However, the features provided by the e-pSyLon platform are appropriate for carrying out most of the synchronous e-training activities of a large corporation.

5 Validation

The efficient use of network resources enables two scenarios: activities with a very large number of participants and low interactivity, and activities with a relatively low number of participants but with a high level of interactivity. The e-pSyLon e-training platform has been tested in these two scenarios. Initial tests were conducted to verify the scalability of the platform to a large number of users. Finally, the platform was also tested in a real highly interactive e-training activity to gather the opinions of the participants.

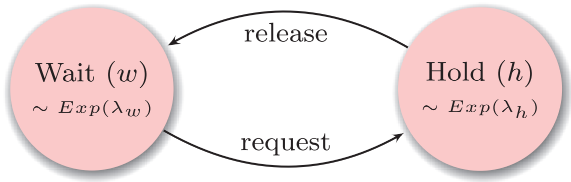
The scalability of the platform mostly depends on the number of data streams flowing through the network. Table 2 depicts the network bandwidth required by a single stream of each feature of the e-training platform. The number of data streams flowing depends on the number of sources and multicast islands. The scalability of the relay mesh is analyzed in [22]. The main conclusion of the analysis is that the dispersion of participants has more impact on scalability than the actual number of participants.

The platform uses a single RP and a single BFCP server. This could lead to scalability issues, as they might become bottlenecks of the platform. However, only in rare occasions is the scalability of the platform limited by the RP. The RP acts exclusively as an entry point to the platform so the workload supported by the RP during an activity is minimal. An excessive workload in the RP may only affect joining and leaving times of participants.

Participants in an e-training activity should hold a floor to interact with the rest of the participants using one of the features of the platform. Thus, they should contact the BFCP server to request or release floors. The BFCP server might be a bottleneck depending on the interactive level of the e-training activity and the number of participants.

Table 2 Network bandwidth used by a data stream

Feature	Bandwidth (kbps)
Video	52.8 (160x120x5 fps) − 308.4 (320x240x15 fps)
Audio	82.7(G.711), 23.35 − 47.8(Speex), 28.8 and 38.4(iLBC)
Whiteboard	$<$ 1(Navigation), $<$ 1(Annotations), 5.0(Telepointer)
Instant messaging	$<$ 1



Interactivity	λ_w	λ_h
Low	1/60	1/20
Medium	1/30	1/15
High	1/10	1/10

Fig. 12 Operation of the BFCP injector and parameters used in the tests

Tests were carried out to analyze the scalability of the BFCP server. A BFCP traffic injector was used to emulate participants, who requested a floor and released it after a holding time (h). The emulated participant requested the floor again after a waiting time (w). These times were shorter for a highly interactive activity and longer for an activity with low interactivity. The binary format of the messages interchanged between the BFCP server and the injectors is described in [9]. Figure 12 shows the operation of the BFCP injector and the three interactive levels defined for the tests. The waiting and holding times are random variables with exponential distributions.

Participants can subscribe to floors, so the BFCP notifies them of all the participants' requests. The BFCP server only notifies the requesting participant when participants are not subscribed to floors. Both scenarios were deployed in the tests.

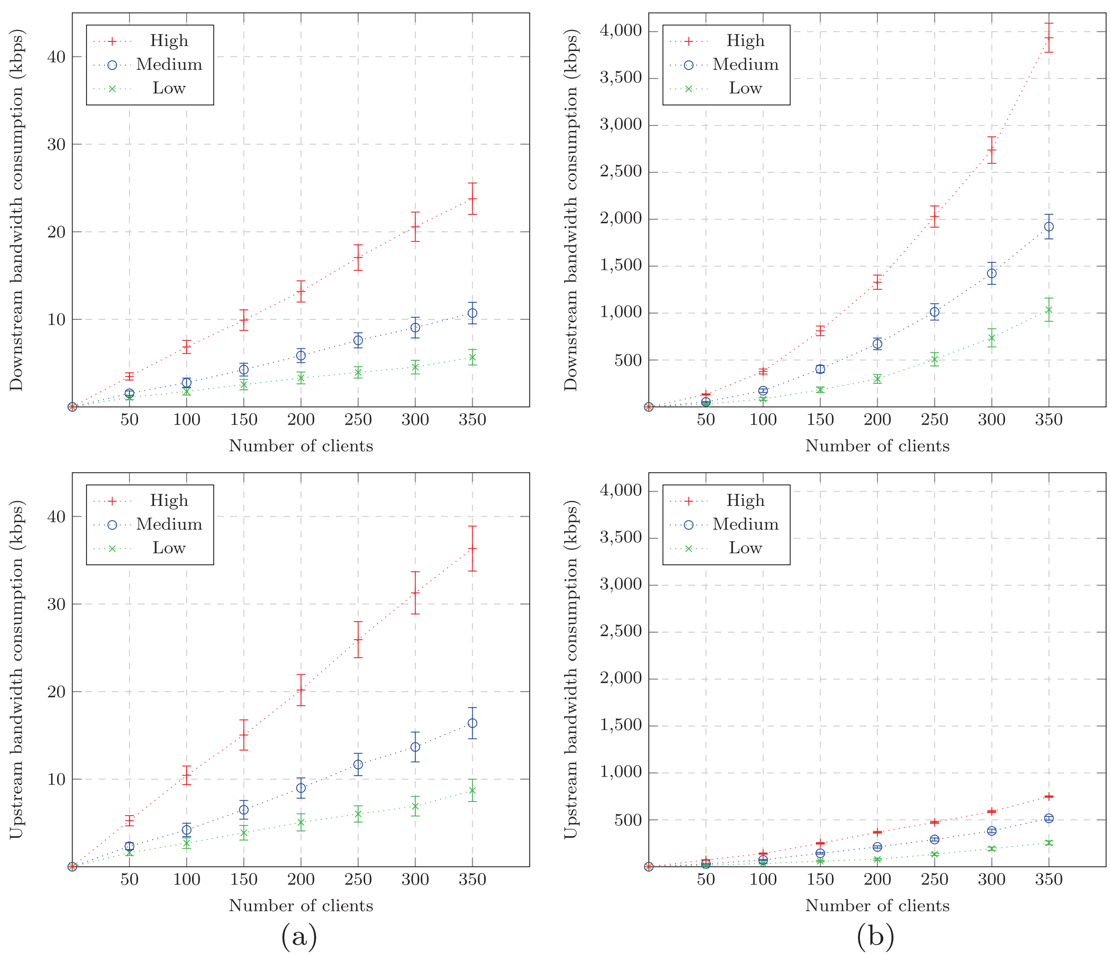


Fig. 13 Network resources used by the BFCP server depending on the interactive level when participants are not subscribed to floors (**a**), participants are subscribed to floors (**b**)

Figure 13 shows the network resources used at the BFCP server when participants are subscribed to floor notifications and when they are not subscribed. The confidence intervals for $\alpha = 0.05$ are also plotted. The BFCP can easily support 350 participants in a highly interactive activity with a low bandwidth when participants are not subscribed to floors. The number of BFCP messages sent by the server to participants grows exponentially when participants subscribe to floors. Thus, the subscription of participants to floors might affect scalability in very large activities. Nevertheless, the subscription of participants to floor status changes is not critical for large activities. A participant who is not subscribed to a floor is informed when he is granted or revoked the floor, although he is not informed when other participants are granted or revoked. Therefore, the scalability of the whole platform depends on the scalability of the relay mesh as explained in [22], which depends on the number and dispersion of participants. The BFCP server will never become a bottleneck during an e-training activity.

The e-pSyLon platform was also tested in a real training activity in ArcelorMittal, a scenario with high interactivity among a small number of participants. ArcelorMittal is the world's leading steel manufacturer, with presence in more than 60 countries. "La Toba" is the training center of ArcelorMittal Spain, which develops the training program of the corporation in Spain and promotes almost 3,000 training activities per year, involving more than 19,000 learners and some 500 instructors. In 2011, more than 3,800 training activities were held before the end of June.

The e-pSyLon platform was tested in a course on electrical risks prevention. The activity was conducted over a single working day, and was planned for two hours involving eight employees of ArcelorMittal. The average age of the participants in the activity was 33.5, and the male-female ratio was 7–1. None of them was affected by physical impairments. The participants were spread over geographical locations across the country, as shown in Fig. 14. The instructor was located in Avilés, one

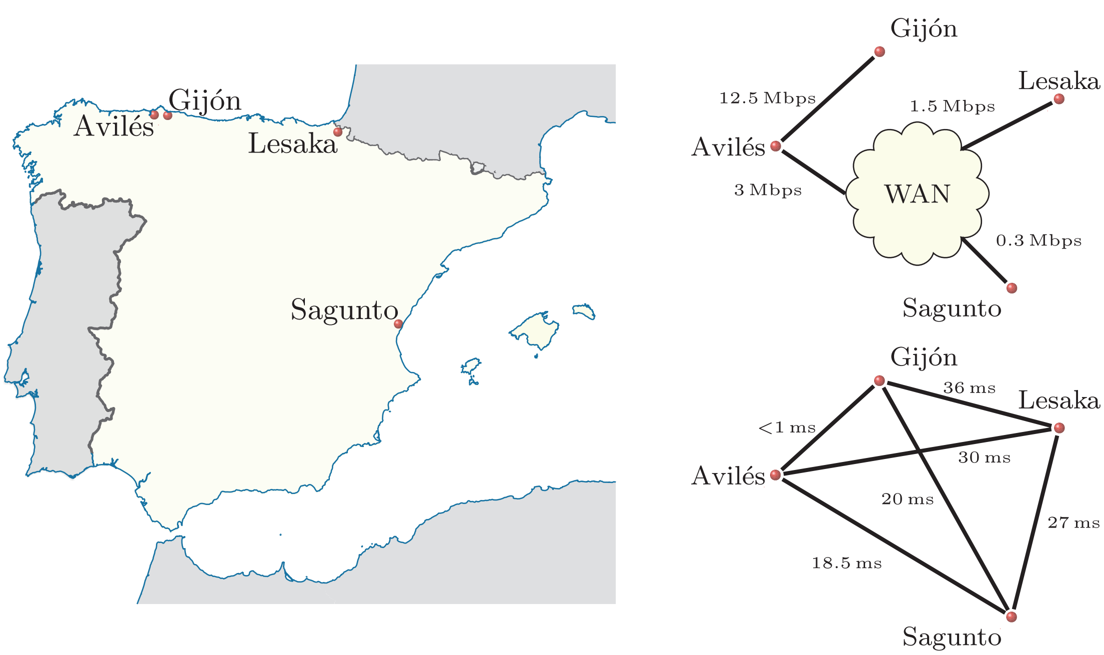


Fig. 14 Geographical locations of participants in the case study, available network bandwidth and observed network latency

learner was in Gijón, three in Lesaka, and the rest in Sagunto. The figure also shows the bandwidth available in the network links of the sites for carrying out synchronous e-training activities and the latency observed between sites computed as half of the round-trip time.

The course is regularly conducted in the training center, so the benefits from the use of the platform to carry out the course remotely can be inferred comparing the satisfaction of users participating locally and remotely. This case study is not intended as an exhaustive analysis of the application of a synchronous e-training tool, but as a preliminary indication of the benefits and the limitations of the e-pSyLon e-training platform when applied to a real e-training activity.

Data about participants were collected through a pre-survey. This pre-survey provided information about participants' experiences in e-training courses using synchronous and asynchronous e-learning tools, as well as their willingness to participate in this synchronous activity. Furthermore, their skills in working with computers were also evaluated.

All the participants in the activity completed the pre-survey. A five-point Likert scale was used to gather the opinion of users. All the participants use computers every weekday in their workplace and at home, while 75% of them use computers every day. In their opinion, they have good computer skills (3.85 on average), except the instructor who considered that his computer skills were medium (3). None of them had previously participated in a synchronous or asynchronous e-training activity, and they expressed reservations about their effectiveness. The participants did have previous experience using synchronous tools such as videoconferencing and instant messaging tools.

Just before the e-training activity, all the participants were enrolled in a brief introduction to the use of the e-pSyLon client tool. This introduction covered the basics of the joining e-training activities and interacting with other participants. Also, a manual of the client tool was published as a quick reference of the available functionality.[1]

The instructor acted as the moderator of the activity granting and revoking all the floors. All participants could communicate with the others using any of the available features provided by the platform, except those from Sagunto, who were not allowed to send or receive video from the rest of the participants. Although the bandwidth of the access link of Sagunto (8 Mbps) is high compared to that available in the access link of Lesaka (2 Mbps), the regular traffic of the corporation is significant higher in Sagunto, so the free available bandwidth in the network link of the site is low, as illustrated in Fig. 14. Thus, the resources consumption in the network link of Sagunto was reduced by forbidding Sagunto learners to use the video channel.

The floor policies of the activity established a maximum of 2 holders of the video floor, so the video channel could be used by none, one or two participants simultaneously anytime. The limit for the audio channel was 3 simultaneous participants to limit the use of network bandwidth. All participants were allowed to use the instant messaging feature by default in order to pose questions or request the attention of the instructor. All participants used iLBC audio codec with a packet size of 20 ms and a network consumption of 38.4 kbps per stream, and VC-1 video codec with a

[1] http://www.atc.uniovi.es/etipo/manual/.

resolution of 160 × 120 and 15 frames per second, resulting in a network consumption of 106.96 kbps per stream.

The activity lasted 30 extra minutes due to some initial problems with the configurations of the audio devices. After solving these issues, the instructor started the e-training activity. The e-pSyLon platform logs provided an indication of the interaction between participants during the activity. The interaction between participants was low at the beginning of the activity (only the instructor and one learner used the audio and video channels), but increased as the activity progressed. Initially, the learners preferred to ask questions using instant messages (15 questions), but eventually they felt comfortable using both the audio and video channels. Fifteen floor requests were granted for the video channel, while 20 requests were granted for the audio channel. This does not mean that only 20 feedback events were provided by the learners, as the instructor preferred to keep the audio floor granted to the maximum number of learners instead of granting and revoking the floor continuously. Only when the maximum number of audio floor holders were reached and another learner required the instructor's attention did the instructor revoke the floor from one of the holders and grant it to the requesting learner. The instructor required the learners to complete some exercises by using the annotation feature. The learners completed the exercises easily. At the end of the activity, all the participants were requested to complete a final questionnaire.

Table 3 shows most of the questions of the final questionnaire. All the questions used a five-point Likert scale. The control column is the average score for the last three on-campus activities on the same course carried out in the training center. Twenty-six participants were enrolled in the on-campus activities. These participants were different from those enrolled in the e-training activity. The e-pSyLon column represents the score for the activity carried out using the e-pSyLon platform. Three open-ended questions (9 to 11) were also included in the questionnaire to solicit opinions about the benefits of using e-pSyLon and any other pertinent comments.

In general, the training experience was successful, although a few problems were reported, especially related with the audio quality. These problems were caused by peaks in the network bandwidth consumption of some sites, as the activity had to share the bandwidth resources with other traffic of the corporation. This was especially noticeable since the activity was carried out during office hours. However, the number of incidents was low.

Table 3 Final questionnaire

#	Question	Control	e-pSyLon
1	I think the activity was organized properly	3.6	3.7
2	I think the duration of the activity was adequate	4.1	3.5
3	I think the way the activity was taught eased learning	4.3	4.5
4	I think my doubts were all solved	3.7	3.4
5	I think the instructor fostered participation during the activity	3.8	4.1
6	I'm satisfied with e-pSyLon while interacting with other participants	–	3.9
7	I would like to use e-pSyLon in future activities	–	4.1
8	Overall satisfaction	4.4	4.0
9	What are the advantages of using e-pSyLon?		
10	What aspects of e-pSyLon can be improved?		
11	Any additional comments?		

Both the instructor and the learners expressed their satisfaction with the use of the tool. Participants' reactions and comments provided important insight about the benefits and limitations of the e-training tool and the e-training activity. All of them agreed on the suitability of the tool for carrying out e-training activities, although they pointed out the disturbance produced by network issues.

Participants consider that the duration is more adequate for the on-campus activity (4.1) than for the online activity (3.5). This can be partially explained by the extra 30 min at the beginning of the online activity to configure the audio devices properly. The learners also think that the learning was easier in the online activity, mainly because of the additional information provided by the instructor using annotations on the training material. Similarly, the learners consider that the number of interactions were slightly higher during the e-training activity (4.1) in comparison with the control activity (3.8). Finally, the overall satisfaction of participants is lower in the e-pSyLon activity. The participants consider that audio glitches make interactions difficult. However, the overall satisfaction is high (4).

6 Conclusions and future work

A synchronous e-learning client tool and its application to e-training activities in a large corporation have been presented. This tool operates in a synchronous e-training platform that takes into account the specific characteristics of corporate networks to make efficient use of network resources using IP multicast where available. Thus, the e-training platform can be used to carry out low-interactive synchronous activities with a high number of users and highly interactive activities with a small number of users. Unlike commercial e-learning solutions, the whole platform is built upon open standards, so interoperability with other applications is guaranteed.

The e-training platform is composed of various entities which inter-operate during an e-training activity. The RP uses SIP to authorize users to join activities and negotiate their multimedia configuration. The floor control server develops the floor control policies of the activities, allowing the instructor to control interactions among participants, while various RTP relays can be used to take advantage of the multicast delivery available in the IP multicast islands of the corporation.

The client tool has a layered design, so additional features can be added. The client tool provides multipoint audio and video communication features, instant messaging and presence control, shared whiteboards with annotations and telepointers, and floor control for the management of all the features of the platform.

Scalability tests proved that the platform scales to a high number of users easily. A synchronous e-training activity was also carried out using the platform to evaluate the satisfaction of users while using the e-pSyLon client tool. Although users reported some inconveniences due to the limited available network bandwidth of some corporate sites, they agreed on the suitability of the tool for carrying out e-training activities. In general, all of them found the functionality offered by the tool appropriate, bringing them new possibilities of interaction.

Future work will be focused on extending the use of the e-pSyLon platform in the ArcelorMittal training plan. An in depth study into the benefits of the use of the e-pSyLon platform in synchronous e-training activities is planned. The additional communication capabilities of the platform should prove better than the learning experience using traditional asynchronous learning platforms. Blended experiences combining synchronous and asynchronous tools offer new possibilities that will also be studied.

Acknowledgements This work was partially funded by ArcelorMittal under contract number CN-06-038. Financial support (grant:UNOV-10-BECDOC-S) given by the University of Oviedo-Banco de Santander is acknowledged. This work was partially funded by the Spanish National Plan of Research, Development and Innovation under the projects TIN2008-06045-C02 and TIN2011-24903.

References

1. Alavi M (1994) Computer-mediated collaborative learning: an empirical evaluation. Manag Inf Syst Q 18(2):159–174
2. Almeroth KC (2000) Evolution of multicast: from the MBone to interdomain multicast to internet2 deployment. IEEE Netw 14(1):10–20
3. Barnes M, Boulton C, Levin O (2008) A framework for centralized conferencing. RFC 5239, Internet Engineering Task Force
4. Bijlani K, Rangan P, Subramanian S, Vijayan V, Jayahari K (2010) A-view: adaptive bandwidth for telepresence and large user sets in live distance education. In: Proceedings of the 2nd international conference on education technology and computer (ICETC 2010). Shanghai, China, pp V2219–V2222
5. Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W (1998) An architecture for differentiated services. RFC 2475, Internet Engineering Task Force
6. Bolot JC, Vega-Garcia A (1996) Control mechanisms for packet audio in the internet. In: Proceedings of the 15th annual joint conference of the IEEE computer societies (INFOCOM 1996), vol 1. San Francisco, CA, USA, pp 232–239
7. Boulos MNK, Taylor AD, Breton A (2005) A synchronous communication experiment within an online distance learning program: a case study. Telemed J e-Health 11(5):583–593
8. Bouras C, Gkamas A, Kapoulas V, Stamos K (2002) Desktop synchronous distance learning application enhanced with efficient chair control capabilities. In: Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA 2002). Las Vegas, NV, USA, pp 1158–1164
9. Camarillo G, Ott J, Drage K (2006) The binary floor control protocol (BFCP). RFC 4582, Internet Engineering Task Force
10. Casner S, Deering S (1992) First IETF internet audiocast. ConneXions 6(6):10–17
11. Chassie K (2002) The allure of e-learning. IEEE Potentials 21(3):33–35
12. Civanlar M (2000) RTP payload format for real-time pointers. RFC 2862, Internet Engineering Task Force
13. Deshpande S, Hwang JN (2001) A real-time interactive virtual classroom multimedia distance learning system. IEEE Trans Multimedia 3(4):432–444
14. El Saddik A, Yang D, Georganas N (2007) Tools for transparent synchronous collaborative environments. Multimed Tools Appl 33(2):217–240
15. El Saddik A, Rahman A, Abdala S, Solomon B (2008) PECOLE: P2P multimedia collaborative environment. Multimed Tools Appl 39(3):353–377
16. Emswiler T (1995) Using the multicast backbone (MBone) for distance learning: a case study. Master's thesis, Naval Postgraduate School, Monterey, California
17. Floyd S, Jacobson V, Liu CG, McCanne S, Zhang L (1997) A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM Trans Netw 5(6):784–803
18. Fortino G, Nigro L, Pupo F (2001) An mbone-based on-demand system for cooperative off-line learning. In: Proceedings of the 27th euromicro conference. Warsaw, Poland, pp 336–344
19. Fung A, Ledesma J (2005) Extending the classroom. In: Proceedings of the 6th international working conference on information technology in educational management (ITEM 2004). Las Palmas de Gran Canaria, Spain, pp 47–56

20. García DF, Uría C, Granda JC, Suárez FJ, González F (2007) A functional evaluation of the commercial platforms and tools for synchronous distance e-learning. Int J Educ Inf Technol 1(2):95–104
21. Granda J, Uría C, García D, Suárez F, González F (2008) Design issues in remote visualization of information in interactive multimedia e learning systems. In: Proceedings of the IEEE international conference visualisation (VIS 2008). London, UK, pp 70–76
22. Granda JC, García DF, Nuño P, Suárez FJ (2010) An efficient networking technique for synchronous e-learning platforms in corporate environments. Comput Commun 33(14):1752–1766
23. Hellstrom G, Jones P (2005) RTP payload for text conversation. RFC 4103, Internet Engineering Task Force
24. Higuchi Y, Mitsuishi T, Go K (2006) An interactive multimedia instruction system: IMPRESSION for multipoint synchronous online classroom environment. In: Proceedings of the 10th international conference on knowledge-based intelligent information and engineering systems (KES 2006), vol 4252 LNAI - II. Bournemouth, pp 1027–1034
25. Isaacs E, Tang J (1994) What video can and cannot do for collaboration: a case study. Multimed Syst 2(2):63–73
26. Isenhour P, Carroll J, Neale D, Rosson M, Dunlap D (2000) The virtual school: an integrated collaborative environment for the classroom. Educ Technol Soc 3(3):74–86
27. ITU-T (1998a) Multipoint still image and annotation protocol. Recommendation T.126, International Telecommunication Union
28. ITU-T (1998b) Protocol for multimedia application text conversation. Recommendation T.140, International Telecommunication Union
29. Jacobson V, McCanne S (1993) Visual audio tool. Lawrence Berkeley Laboratory. ftp://ftp.ee.lbl.gov/conferencing/vat
30. Johnson DW, Johnson RT, Smith KA (1991) Cooperative learning: increasing college faculty instructional productivity. ASHE ERIC Higher Education Report 4, The George Washington University, School of Education & Human Development, Washington, D.C.
31. Latchman H, Salzmann C, Gillet D, Bouzekri H (1999) Information technology enhanced learning in distance and conventional education. IEEE Trans Ed 42(4):247–254
32. Latchman H, Salzmann C, Gillet D, Kim J (2001) Learning on demand - a hybrid synchronous/asynchronous approach. IEEE Trans Ed 44(2):208
33. Lecce VD, Giove A, Quarto A (2009) A virtual classroom interface for student participation measurement. In: Proceedings of the international conference on virtual environments, human-computer interfaces and measurements systems (VECIMS 2009). Hong Kong, China, pp 255–260
34. Ludlow BL, Duff MC (1996) The instructor-producer relationship: a partnership for effective distance education. Tech. Rep. RC 020 572, In: Rural goals 2000: building programs that work
35. Malpani R, Rowe LA (1997) Floor control for large-scale MBone seminars. In: Proceedings of the 5th ACM international multimedia conference. Seattle, WA, USA, pp 155–163
36. Marjanovic O (1999) Learning and teaching in a synchronous collaborative environment. J Comput Assist Learn 15(2):129–138
37. McCanne S, Jacobson V (1995) vic: a flexible framework for packet video. In: Proceedings of the ACM multimedia. San Francisco, CA, USA, pp 511–522
38. Nuño P, Granda JC, García DF, Suárez FJ (2010) Automatic deployment of a communication mesh for synchronous e-learning activities. In: Proceedings of the 5th international conference on systems and networks communications (ICSNC 2010). Nice, France, pp 213–220
39. Pahud M (2008) ConferenceXP research platform: toward an extensible collaborative environment. Microsoft research. http://research.microsoft.com/conferencexp
40. Papert S, Harel I (1991) Constructionism. Ablex Publishing Corporation, New York, NY, USA
41. Paxson V (1999) End-to-end internet packet dynamics. IEEE/ACM Trans Netw 7(3):277–292
42. Premchaiswadi W, Tungkasthan A, Jongsawat N (2010) Enhancing learning systems by using virtual interactive classrooms and web-based collaborative work. In: Proceedings of the IEEE eduation engineering conference (EDUCON 2010). Madrid, Spain, pp 1531–1537
43. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E (2002) SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force
44. Schulzrinne H, Casner S, Frederick R, Jacobson V (2003) RTP: a transport protocol for real-time applications. RFC 3550, Internet Engineering Task Force
45. Snow C, Pullen J, McAndrews P (2005) Network educationWare: an open-source web-based system for synchronous distance education. IEEE Trans Ed 48(4):705–712

46. Ullrich C, Shen R, Tong R, Tan X (2010) A mobile live video learning system for large-scale learning–system design and evaluation. IEEE Trans Learning Technol 3(1):6–17
47. Weller M (2007) The distance from isolation. why communities are the logical conclusion in e-learning. Comput Educ 49(2):148–159
48. Yang Z, Liu Q (2007) Research and development of web-based virtual online classroom. Comput Educ 48(2):171–184
49. Zhao X, Zhang Y (2006) An instructor-oriented prototype system for virtual classroom. In: Proceedings of the 6th international conference on advanced learning technologies (ICALT 2006). Kerkrade, The Netherlands, pp 200–204
50. Zhao X, Zhong Y, Matsumoto M (2006) A real-time interactive shared system for distance learning. In: Proceedings of the 12th international multi-media modelling conference (MMM 2006). Beijing, China, pp 102–107

Juan C. Granda is an associate professor in the Department of Computer Science and Engineering at the University of Oviedo. He received his M.S. and Ph.D. degrees in computer science from the University of Oviedo in 2004 and 2008, respectively. In recent years, he has been working on several projects related to interactive multimedia services. His research interests include synchronous e-learning systems and multimedia networking.

Pelayo Nuño received his M.S. degree in computer science from the University of Oviedo in 2009, where he is currently working on his Ph.D. degree in computer science. He has been involved in several projects within the Department of Computer Science and Engineering at the University of Oviedo. His research interests include multimedia networking and synchronous e-learning systems.

Francisco J. Suárez is a professor in the Department of Computer Science and Engineering at the University of Oviedo and received his Ph.D. degree from that University in 1998. His research focuses on the evaluation of networked multimedia systems and he has been recently working on several projects related to interactive multimedia services.

María A. Pérez received his M.S. degree in mechanical engineering in the Univeristy of Oviedo in 2004. She is currently a training coordinator in ArcelorMittal Spain. She has been actively involved in the development of new ways of training, such as multimedia tools and simulators.

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.

Jan. 7, 2013

Dear Dr. Juan Carlos Granda,

It is my pleasure to inform you the acceptance of your manuscript entitled:

SYNCHRONOUS E-LEARNING TOOLS

For publication in the forthcoming editing book entitled:

E-Learning: New Technology, Applications and Future Trends

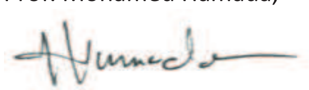
To be published by NOVA Science Publication (www.novapublishers.com) during 2013.

Thank you for your participation by your research in the book.

Best regards,

The Book Editor:

Prof. Mohamed Hamada,



The University of Aizu,
Aizuwakamatsu, Fukushima,
JAPAN.
E-mail: hamada@u-aizu.ac.jp
Tel: +81 242 372617
Fax: +81 242 372735

Chapter 6

SYNCHRONOUS E-LEARNING TOOLS

Juan C. Granda,* Pelayo Nuño, Francisco J. Suárez, Daniel F. García
Department of Informatics, University of Oviedo

Keywords: Synchronous e-learning, real-time communications, application features.

Abstract

Synchronous e-learning has become a very successful learning modality. Synchronous e-learning tools include a wide variety of features enabling real-time interactions among learners and instructors. These interactions emulate the real-time interactions that occur in face-to-face traditional learning. This chapter introduces the basic features of synchronous e-training tools. An analysis of the prominent synchronous e-learning tools is carried out, concluding that most of the tools include audio and videoconferencing, document sharing and instant messaging features. Other features such as dynamic polls or the capability to divide learners into groups are less common. Furthermore, hardware solutions for synchronous e-learning and collaboration, and virtual immersive environments are described.

1. Introduction

E-learning is becoming increasingly popular as a tool for both education in academic institutions and the development of human resource training plans in large enterprises. The

*E-mail address: jcgranda@uniovi.es

introduction of e-learning in the curriculum design provides advantages such as an increase of the availability and flexibility of the learning process, learners' autonomy, lower costs due to the reduction of displacements, and a long-term increase of the productivity. There are also some drawbacks that must be considered: the learning curve in the case of users with low computer skills, the unavailability of the instructor in some modalities, the required commitment of learners and the initial rejection due to the lack of previous experiences both in educational and business processes.

E-learning can be classified into two types according to the restrictions imposed on the learning process: asynchronous and synchronous e-learning. Learners can participate in the learning process anywhere anytime in asynchronous e-learning. In contrast, learners must join synchronous e-learning activities at the same time in order for the learning process to take place. Synchronous e-learning has multiple advantages over other instructional strategies. This is due to the especial suitability to engage dispersed learners preserving similarities to traditional on-campus learning by emulating face-to-face interactions in virtual classrooms. Interactive communication features such as audio and videoconferencing, or shared whiteboards and applications, translate the traditional face-to-face learning paradigm to an online environment. Unlike asynchronous e-learning, synchronous e-learning makes real-time interactions among learners possible, which promote collaborative learning, avoiding anxieties suffered by learners derived from the absence of an instructor during the learning experience.

From the analysis of several human and technological challenges faced by synchronous e-learning it is possible to extract the pedagogical principles and technical requirements that should be accomplished by a synchronous e-learning tool. A full catalog of features can be enumerated according to such principles and requirements. However, this catalog of features may vary depending on the purpose, the constraints, and the desired interactivity for the learning process.

This chapter widely covers the aspects, principles, requirements and features related to synchronous e-learning tools. A catalog of features available for instructors and learners is introduced, focusing on functionality rather than technological background. Furthermore, a survey of the prominent synchronous e-learning tools is included presenting a detailed functional evaluation.

2. Features of Synchronous E-learning Tools

One of the requirements of a synchronous e-learning tool is to deliver the speech and comments of the instructor to learners, facilitating the acquisition of the competences and skills being trained during the synchronous e-training activity by learners. Furthermore, a synchronous e-learning tool must allow for interactive communications so the instruct can adapt the pace of the activity according to the feedback received from learners. The learning process can be supported by various multimedia features. These features are introduced below, highlighting their utility and importance in synchronous e-training activities. Some of the features provided by synchronous e-learning tools can be used during synchronous activities, although other features can be used before and after the activities.

2.1. Before and After the Activity

Instructor and learners must usually register on a website where they can join virtual classrooms. In this website the instructor can manage the scheduling of activities, authorized users and passwords, learning material, recording of activities, etc. The instructor can also send an e-mail to learners reporting the schedule of the activity or last-minute information about some aspects to analyze during the activity. These e-mails can attach a link to the virtual classroom in those synchronous e-learning tools where users must join activities using a website. Thus, learners are able to access to the class on the fly without installing any software in their computers. Similarly, these links may be available on the website once the instructor has scheduled the class. Obviously, learners have less options available than the instructor in the website. They can confirm their attendance to activities, check schedules and access previously recorded synchronous activities.

Once a synchronous activity is up, the instructor can decide whether to share the recorded activity and any other material generated with learners. It should be noted that synchronous e-learning is not rigid but it is quite usual to complement synchronous e-learning tools with asynchronous learning features. For example, the instructor often provides supplementary material for learners in a repository, usually located in a website of a Learning Management System (LMS). This repository is different from that used for the essential material of the activity. Furthermore, it is also possible to open discussion forums where learners can discuss about the learning experience so they can clarify doubts with other learners and the instructor after activities.

Besides the integration with e-mail clients (*i.e.* Microsoft Outlook) and personal organizers (*i.e.* Lotus Notes), the websites of the synchronous e-training tools can also be integrated with directory services such as those based on the Lightweight Directory Access Protocol (LDAP). This allows for validating and autheticating instructors and learners from the records of a directory service and managing their user profiles transparently. There is no need for an independent profile management for the synchronous e-learning tool.

2.2. Role of the Instructor

Although not a feature in itself, the role of the instructor, teacher or presenter is critical in a synchronous e-learning activity. Conceptually, the instructor is another participant in the activity, but with more privileges and functionalities than the rest. Besides having more number of functionalities available than other users, the instructor is able to grant or revoke privileges to learners, such as allowing annotations or participation in the activity, either through the chat or audio and videoconference. The instructor can also grant a learner the same privileges as the former, enabling a scenario where there are several instructors. In any case, the initial instructor never loses control over the activity. The instructor can also manage specific functionalities related to learners both individually and collectively. For instance, the instructor can give learners the permission to change the layout of the windows of the synchronous e-learning tool on their screens, or he can force them a specific layout as in the case of contents that need to be presented in full screen. Some synchronous e-learning tools allow the instructor to monitor the learners's work by spying what it is displayed on their screen.

In summary, the instructor has a number of privileges for managing the synchronous activity. Sometimes these privileges can be used to deal with connectivity issues among learners, such as a situation in which several learners are not following the activity properly due to network congestion. In this case, the instructor can disable the video and the audio of these learners so they can participate in the activity although with a reduced functionality.

2.3. Presence Control

From the point of view of learners, the presence control only gives the ability to check who is attending or connected to the activity. However, presence control is absolutely essential in order to track learners in synchronous e-learning activities. The knowledge about learners attending the activity is critical for a wide range of features. For example, users should know who is the sender of a text message or who is writing on the shared whiteboard. Therefore, most of the synchronous e-learning tools show a list of the participants attending the activity on the screen using icons to identify their state (instructor, learner, absent, speaking, privileges granted, etc.). Finally, some tools allow participants that are not attending the activity to chat with those learners that are within the activity. These participants are usually referred to as ghost users.

2.4. Instant Messaging

This is a simple but essential feature allowing communication between participants even under severe network conditions, since the required network bandwidth to deliver text messages is negligible. This feature offers the same options than typical instant messaging applications: chat among participants, chat privately with one or more participants, customize text font, etc. Instant messaging allows learners to pose questions to the instructor without disrupting the activity as occurs with oral questions. Furthermore, instant messaging is also very useful to provide feedback from learners to the instructor and to enable interactions among learners. The instant messaging window is usually located close to the attendance list (presence control) on the screen.

2.5. Audioconferencing

Almost all synchronous e-learning tools support audioconferencing. Audioconferences among participants can be public, or sidebar conversations between two or more participants can be established. Audio streams require a significant amount of network resources but less than video streams, although the importance of the former is greater than the latter. Audio quality must be as high as possible in synchronous e-learning activities since the instructor's speech and explanations must be clearly comprehensible. Thus, audio data must be prioritized over other kinds of data, so adverse network conditions do not affect audio comprehension.

Each participant in an audioconference generates an audio stream that must be sent to the rest of participants. Therefore, both upstream bandwidth from the sender's connection and downstream bandwidth from the connections of the rest of participants are consumed. This implies that the maximum number of participants concurrently using the audioconference feature is often limited. A floor control mechanism is required in order to share the audio channel among participants, avoiding the overlapping of audio streams from various participants.

When the audio delivery is based on multicast communications an audio stream is not replicated for every receiver, which allows for an increase of the number of participants in the audioconference. Multipoint audioconferencing fosters collaboration, since real-time audio communications are possible between many simultaneous participants. However, the number of simultaneous audio streams should be limited to avoid many participants talking at the same time.

2.6. Videoconferencing

Videoconferencing is the most resource-consuming feature. The video stream from the instructor usually portrays the instructor's talking head. Therefore, it does not provide any pedagogical content, yet the video stream reinforces the sensation of the presence of the instructor, and avoids a sense of isolation in the learners. Multipoint videoconferencing can be used to emulate face-to-face interaction between participants in synchronous e-learning activities. In fact, non-verbal communication is possible, although video streams must have an appropriate temporal resolution. The video stream from the instructor can also be used to illustrate the steps of a manual process or show last-minute information, such as printed documents, pictures, mechanical pieces, etc. In this situation, both the temporal and spatial resolutions of the video stream must be high, so the network bandwidth requirements will also be high.

The number of simultaneous video streams is limited in all the synchronous e-learning tools due to their high bandwidth requirements. When the underlying network has low-bandwidth links, only one participant, usually the instructor, is allowed to use the video channel. The appearance of video degradations is the first indication of network issues. When this occurs, deactivating the videoconference is the best solution since a poor video transmission is useless and consumes a network bandwidth that could be used by other functionalities. Similarly to audioconferencing, multicast communication minimizes the amount of data transmitted, so the performance of videoconferencing increases. However, the selected video quality also determines the bandwidth required by video streams. Parameters such as resolution, framerate, or codecs can be customized. A common scenario for videoconferening uses a resolution of 320×240 pixels and 15 frames per second.

2.7. Document sharing

Document sharing (or content presentation) is the most important feature of a synchronous e-learning tool. Most of the synchronous activities are developed using some contents or documents that are presented to the participants. Every e-learning tool has a main area on the screen for representing the contents selected by the instructor. Any participant can also present contents when granted by the instructor. Synchronous e-learning tools support a wide variety of file formats: PowerPoint, Project, Word, Excel, Acrobat (PDF), AutoCAD, images (JPG, GIF, PNG), etc. PowerPoint files are the most widely supported format by e-learning tools. PDF files are also highly supported, so it is recommended to convert any unsupported document format to PDF. There is no need to install the authoring tool for the formats supported by a synchronous e-learning tool. When the format is not supported, the instructor may share the authoring tool with learners, but this imposes that the authoring tool is installed in the computer of the instructor.

The window where the documents are shared can be resized, so it is possible to present documents in full screen while attending the oral speech of the instructor. Each learner can individually manage how the documents are displayed on the screen if desired by the instructor. In this case, a learner may increase or decrease the size of the document on the screen or navigate through the pages forward and backward. Synchronous e-learning tools also include a list of slides as a summary of the document, enabling a quick access to all the sections of the document. This summary is usually a textual scheme, but there are also tools that show a list of thumbnails of the slides. The presented contents can be stored in the instructor computer or in a multimedia server where they are delivered to all the participants. When the instructor decides to share documents not stored in the multimedia server, some time is needed to adapt the contents to be served to the participants. Usually, the documents are also uploaded to the multimedia server during such time interval.

A feature closely related to document sharing is the virtual pointer (*a.k.a.* telepointer), which refers to the cursor pointing device of any participant when it is visible to the rest. A virtual pointer allows a participant to point to a specific region in the content presentation area. This is useful to emphasize a concept in the current slide. For example, a instructor can use a virtual pointer to highlight a question submitted to learners.

2.8. Shared Whiteboard

The shared whiteboard is another important feature implemented in synchronous e-learning tools. It is used to share the learning material between learners and instructor. In fact, the audio channel and the shared whiteboard would be enough to support synchronous e-learning activities. It consists on a graphics editor which lets users to draw shapes, insert pictures and write text on a virtual canvas that is shared by those attending the activity. The shared whiteboard is usually closely related to the document sharing feature.

2.9. Annotations

This feature is widely implemented in synchronous e-learning tools. Annotations allow any participant to write or draw on the shared document or the whiteboard to highlight or correct any element of the presentation. For example, the instructor can annotate the slides in the shared whiteboard to provide extra explanations. Annotations are also useful to communicate last-minute information. Learners can also annotate the shared whiteboard to pose questions to the instructor or to other learners. Handwritten annotations enable collaborative work between participants in e-learning activities, as they can be used to jointly draw diagrams or sketches.

2.10. Shared Desktop and Applications

The ability to share an application or the user desktop is a feature present in almost all synchronous e-learning tools. It allows any participant to share a particular application or the entire desktop, so the rest are able to see what he is doing. Shared content is shown in the display area of the participants similarly to shared documents. The instructor can use this feature to explain the steps to perform an operation, how to use a particular application, or indicate how to configure a device. It is also possible to share a part of the desktop, so any application window or content placed in that area is visible to all the participants.

This feature is implemented using video streaming techniques. Although the resolution of the video showing an application window or the desktop must be high, a very low frame rate is enough to transmit the contents. Therefore, the network bandwidth consumed by videoconferences is higher than the bandwidth consumed by this feature.

Finally, another interesting approach to shared contents is the ability to capture screenshots using hotkeys, and automatically submit the capture to the rest of participants.

2.11. Remote Control

Remote control allows a participant, typically the instructor, to remotely use an application or the desktop of another participant. In addition to viewing, it enables the interaction of the remote user, so the user can control the operation of the application or the desktop. Due to the risks involved with this feature, synchronous e-learning tools usually offer a keyboard shortcut or scape button to cancel the remote control any time.

2.12. Synchronous Navigation

This feature allows the instructor to guide the web navigation of learners during synchronous e-learning activities. When this feature is not present, the instructor may share the browser application with learners. However, this solution is highly bandwidth-consuming, since video streaming technology is used to deliver the contents of the instructor's browser window to learners. Conversely, a more interesting option to perform synchronous navigation is the instructor specifying which web pages the learners should visit, so all learners are viewing the same web page as the instructor simultaneously. This requires a very low network bandwidth, as only navigation commands, such as URLs, are sent from the instructor to learners. Thus, each participant accesses web pages individually.

2.13. File Transfer

This feature allows participants to exchange files with other participants by selecting the files to be sent and one or more receivers from the attendee list. The instructor may use the file transfer feature to deliver last-minute contents to learners. This is useful to distribute contents that are not shareable through the shared whiteboard. The same feature can be achieved by using a multimedia repository, such as an HTTP or an FTP repository, so learners may download learning material that have been previously uploaded by the instructor.

2.14. Personal Annotations

Learners can write text annotations into an area reserved that is not visible to the rest of participants. This feature is rarely included in synchronous e-learning tools. The window for personal annotations is usually small and quite similar to the chat window. Learners can also save these notes for future uses.

2.15. Content Storage

The instructor can record a synchronous activity and upload it to a multimedia server, so it can be accessed later. The recording includes all the actions performed by the instructor (speech, annotations, navigation through the contents, etc.). Learners may also record the activity and save it in their computers. This requires the permission of the instructor. The instructor may also enable the recording of specific content presented during the activity such as web pages, sketches or diagrams on the whiteboard, or slides with annotations.

2.16. Division into Groups

This feature allows the instructor to separate learners into various working groups. Within each group, learners have the same features as in a synchronous activity. Only the instructor has the ability to merge groups. This feature is not widely implemented in synchronous e-learning tools since significant network bandwidth can be required. Each working group can be considered as a different activity depending on how data is delivered. The instructor has simultaneous access to all these groups. Thus, learners interactions using audio and videoconferencing can be multiplied according to the number of groups.

2.17. Questions and Polls

The use of questions and polls gives the instructor the ability to gather the opinion or knowledge of learners. Various types of polls can be used: true or false questions, multiple choice questions, filling gaps, and so on. The instructor decides the maximum time available to learners for answering each question. When the time is up or all the learners have answered the question, the instructor can save the answers from each learner. Similarly, the instructor can analyze statistics about the learners' responses to assess their knowledge. The instructor can also gather the opinion of the learners by opening an anonymous poll, which means that answers will be known but the authors will be uncovered. In this case, the answers are usually available to all participants. The area in which the instructor poses questions and learners answer them is often placed above the instant messaging window.

2.18. Instant Feedback

This feature, also referred as intuitive communication, includes several mechanisms to provide instantaneous feedback between learners and the instructor. These mechanisms usually include buttons to send a predefined message to the instructor, such as raising the hand to make a question, giving a yes/no quick answer, or showing mood states with emoticons, etc.

3. Comparison

In addition to those tools specially focused to synchronous e-learning, synchronous activities can be supported by e-meeting or e-collaboration tools. The features of all these types of tools are converging, although some specific characteristics of synchronous e-learning tools such as questions and polls are not included in the other types of tools.

Table 1 shows the main characteristics and features of the tools that are commonly used to carry out synchronous e-learning activities. The functional characteristics have been organized into several groups.

From left to right, the first group of characteristics describes the features of the tools. Almost all tools include audio and videoconferencing and document sharing features, although multipoint conferencing is not widespread. All the tools include an instant messaging and a presence control features. Shared desktop and applications is another typical feature of these tools. Some of them rely on these features to support the synchronous activity instead of using document sharing. The next group includes the characteristics related to the presentation of contents to learners and the possibilities to make and record annotations over the contents. Finally, the last group includes additional session management characteristics.

4. Hardware-based tools

Although synchronous e-learning activities are mainly supported by software tools, sometimes hardware-based tools can be used. The main disadvantage of hardware-based synchronous e-learning is the high costs of the equipment. Each participant in a synchronous activity needs a special equipment and a software license. Furthermore, the flexibility of software tools is higher than that offered by hardware-based. Three hardware-based tools are analyzed.

Polycom [21] provides video and voice conferencing solutions and offer the appropriate network infrastructure and customized hardware to support them. During a conference, the presenter can show any document to the remote participants. This capability allows using these conferencing solutions for e-learning, although the tools lack from many of the features commonly used in synchronous e-learning.

Tandberg [25] offers similar solutions, but it has established alliances with other companies to develop its solutions. Tandberg conferencing solutions can be integrated with WebEx software, while collaborative equipment can be integrated with IBM, Microsoft and Nortel software. Tandberg has been recently acquired by Cisco.

Cisco [9] provides hardware and software support for web conferencing through MeetingPlace. Cisco has acquired WebEx, so the integration with the software by this company has improved in recent years.

5. Virtual-based Tools

An interesting research trend in synchronous e-learning is the use of virtual tools to organize and disseminate knowledge. Many e-learning platforms have arisen following this

	Features															Contents				Session							
	Document sharing	Audioconference	Multi-point audioconference	Videoconference	Multi-point videoconference	Telepointer	Shared whiteboard	Chat	Presence control	File transfer	Polls, short questions, voting	Shared desktop and applications	Shared screenshots	Remote control	Synchronous navigation	Save screenshots	Different layout modes	Annotations	Save personal annotations	Role of instructor	Several instructors	Group management	Watch student screens	Instantaneous feedback	Server-side session recording	Client-side session recording	Secure communications
WebEx Training Center	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Acrobat Connect Pro.	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●		●
Office Live Meeting	●	●		●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●
Elluminate Live!	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●
Centra Symposium	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
GoToMeeting	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●	●
LearnLink	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Windows Meeting Space							●	●	●	●		●									●						
AT&T Connect	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
SameTime	●			●		●	●	●	●	●	●	●		●	●		●	●		●		●		●	●		●
Intercall	●			●	●	●	●	●	●	●	●	●		●	●		●	●		●				●	●		●
e/pop	●	●	●	●	●	●	●	●	●	●		●	●	●	●		●	●	●	●	●				●		●
Groove	●	●	●	●	●	●	●	●	●	●		●			●	●	●		●								●
MegaMeeting	●	●	●	●	●			●	●	●		●	●	●													●
VoxWire	●	●						●	●		●				●		●			●							●
HotConference	●	●		●		●	●	●	●	●		●		●	●	●	●	●		●							●
PictureTalk		●		●		●	●	●	●	●	●	●	●	●	●	●	●	●		●	●					●	●
WebConference	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●	●		●	●					●	●
iLinc	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●	●	●	●		●	●	●	●	●	●
Dimdim	●	●		●		●	●	●	●			●			●			●	●	●	●		●	●	●		
Yugma	●	●	●	●	●	●	●	●	●	●		●						●	●	●	●			●	●	●	●
WizIQ	●	●	●	●	●	●	●	●	●	●	●	●			●			●		●		●		●	●		●
Vyew	●	●	●	●	●	●	●	●	●		●	●	●		●			●		●	●			●	●		
Blackboard Collaborate	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●		●		●	●	●	●		●

Table 1. Comparison of synchronous e-learning tools

approach. They are referred to as virtual environments [14], 3D workplaces [15] or virtual laboratories [4, 11]. These tools can be understood as interactive environments for creating and conducting simulated experiments [19]. They consist of simulation programs related to specific scientific areas, experimental units called objects that encompass data files, a framework for operating over these objects, and tutorials. Using such virtual environments provides benefits such as the reduction of delivery costs, the execution of more effective training processes in less time, more resource utilization and agility, and the ability to measure the training process. They are widely used in several scientific areas [10], but especially in engineering [7, 20], chemistry [23, 26] and medicine [2, 28].

Recently, virtual environments have been replaced by immersive virtual worlds such as *Second Life*, encompassing both educational activities and social interactions [12]. Although virtual worlds are based on the same concept, that is emulation, they promote presentiality, cooperation, membership and so on. Second Life users interact with each other through customizable virtual objects known as avatars. An avatar serves as an alter ego of the participants, allowing for exploring the virtual world, creating stuff and places, participating in virtual lectures and colloquiums, meeting and socializing with partners sharing similar interests and collaborating with them with the same purposes. The main benefit from the use of such virtual objects is that they enforce the presence sense while enhancing both the communication between users and the global awareness of a group.

There are several benefits with the use of these learning mechanisms [5, 29]. Firstly, simulating experiences within a 3D virtual world promotes multi-sensory stimulations which are very useful to acquire and maintain knowledge. Furthermore, the friendly and attractive environment leads to increase cooperation and communication, allowing for a more smooth learning process. Secondly, due to such extremely life-like environment, virtual worlds promote users to train personal skills and probe new ideas applicable to the real world. Finally, these virtual words are flexible since they lack of temporal and spatial restrictions. In addition, they have the ability to emulate unsafe environments becoming them in safe workplaces by avoiding risks related to the management of expensive or dangerous stuff.

Concretely, Second Life has been used in the last decade as a synchronous e-learning tool in many universities [5, 8, 16, 18], especially in nursing and medicine environments [1, 3, 6, 13, 22, 24], organizations [27] and companies across the world [17].

6. Conclusion

Synchronous e-learning has become a very successful learning modality. Blended strategies combining synchronous and asynchronous e-learning result in an enhaced learning experience.

Synchronous e-learning tools are complex applications that include a large set of features. Among others, audio and videoconferencing, document sharing with collaborative annotations foster real-time interactions between learners and instructors. These interactions emulate face-to-face interactions that occurs in on-campus learning.

Most of the synchronous e-learning tools offer a common set of features. Audio and videoconferencing, document sharing and instant messaging are widespread. Other features such as dynamic polls or the capability to divide learners into groups are less common.

However, a convergence is taking place between synchronous e-learning tools and other collaborative tools in the last years.

A higher convergence and integration among synchronous e-training tools, LMSs, e-meeting tools, collaborative tools and personal organizers is expected in the next years.

Acknowledgment

This research has been partially funded by the Spanish National Plan of Research, Development and Innovation under the project TIN2008-06045-C02.

Financial support (grant:UNOV-10-BECDOC-S) given by the University of Oviedo-Banco de Santander is acknowledged.

References

[1] Aebersold, M., Tschannen, D., Stephens, M., Anderson, P., Lei, X., (2011). Second Life: a new strategy in educating nursing students. *Clin. Simul. Nurs.* 1–7.

[2] Aggarwal, R., Moorthy, K., Darzi, A., (2004). Laparoscopic skills training and assessment. *Brit. J. Surg.* 91, 1549–1558.

[3] Ahern, N., Wink, D.M., (2010). Virtual learning environments: Second life. *Nurse Educatr.* 35, 225–227.

[4] Albu, M., Heydt, G., Holbert, K., Albu, M.M., Holbert, K.E., Heydt, G.T., Grigorescu, S.D., (2004). Embedding remote experimentation in power engineering education. *IEEE Trans. Power. Syst.* 19, 139–143.

[5] Beltrán Sierra, L.M., Gutiérrez, R.S., Garzón-Castro, C.L., (2012). Second life as a support element for learning electronic related subjects: A real case. *Com. Edu.* 58, 291–302.

[6] Boulos, M.N.K., Hetherington, L., Wheeler, S., (2007). Second life: an overview of the potential of 3-d virtual worlds in medical and health education. *Health Info. Libr. J.* 24, 233–245.

[7] Casini, M., Prattichizzo, D., Vicino, A., (2003). The automatic control telelab: A user-friendly interface for distance learning. *IEEE. Trans. Educ.* 46, 252–257.

[8] Chung, L.Y., (2012). Virtual reality in college english curriculum: Case study of integrating second life in freshman english course. *Proc. IEEE 26th Int. Conf. Advanced Information Networking and Applications Workshops*, 250–253, Los Alamitos, CA, USA.

[9] Cisco. Meetingplace web conferencing solution. Available: http://www.cisco.com/

[10] Ertugrul, N., (2000). Towards virtual laboratories: A survey of LabVIEW-based teaching/ learning tools and future trends. *Int. J. Elec. Eng. Educ.* 16, 171–180.

[11] Ferrero, A., Salicone, S., Bonora, C., Parmigiani, M., (2003). ReMLab: a java-based remote, didactic measurement laboratory. *IEEE Trans. Instrum. Meas.* 52, 710–715.

[12] Haycock, K., Kemp, J.W., (2008). Immersive Learning Environments in Parallel Universes: Learning through Second Life. *Sch. Libr. Worldw.* 14, 89–97.

[13] Hermanns, M., Kilmon, C., (2011). Second Life as a clinical conference environment: Experience of students and faculty. *Clin. Simul. Nurs.* 08, 297–300.

[14] Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., Hart, P., (1999). VADE: A virtual assembly design environment. *IEEE Comput. Graph. Appl.* 19, 44–50.

[15] Light, J., Miller, J.D., (2002). Miramar: a 3D workplace. *Proc. IEEE Int. Prof. Commu. Conf. (IPCC 2002)* 271 – 282, Portland, Oreg., USA.

[16] Lucia, A.D., Francese, R., Passero, I., Tortora, G., (2009). Development and evaluation of a virtual campus on Second Life: The case of SecondDMI. *Com. Edu.* 52, 220–233.

[17] Ludlow, P., Wallace, M., (2007). The Second Life Herald: The Virtual Tabloid that Witnessed the Dawn of the Metaverse. *The MIT Press*.

[18] Mayrath, M., Sanchez, J., Traphagan, T., Heikes, J., Trivedi, A., (2007). Using second life in an english course: Designing class activities to address learning objectives. *World* 4219–4224.

[19] Mercer, L., Prusinkiewicz, P., Hanan, J., (1990). The concept and design of a virtual laboratory. *Proc. on Graphics Interface* 149–155, Toronto, Ont., Canada.

[20] Overstreet, J.W., Tzes, A., (1999). Internet-based real-time control engineering laboratory. *IEEE Control Syst. Mag.* 19, 19–34.

[21] Polycom. Video and voice conferencing solutions. Available: http://www.polycom.com/

[22] Schmidt, B., Stewart, S., (2009). Implementing the virtual reality learning environment: Second Life, *Nurse Educatr.* 34, 152–155.

[23] Shoichet, B.K., (2004). Virtual screening of chemical libraries. *Nature* 432, 862–865.

[24] Stewart, S., Pope, D., Duncan, D., (2009). Using second life to enhance accel an online accelerated nursing bsn program. *Stud. Health Technol. Inform.* 146, 636–640.

[25] Tandberg. Web conference and collaboration solutions. Available: http://www.tandberg.com/

[26] Tetko, I.V., Gasteiger, J., Todeschini, R., Mauri, A., *et al.*, (2005). Virtual computational chemistry laboratory - design and description. *J. Comput-Aided Mol. Des.* 19, 453–463.

[27] Wang, C.X., Song, H., Stone, D.E., Yan, Q., (2009). Integrating second life into an EFL program in china: Research collaboration across the continents. *TechTrends* 53, 14–19.

[28] Wang, Y., Wang, R., Gao, X., Hong, B., Gao, S., (2006). A practical VEP-based brain-computer interface. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14, 234–239.

[29] Zhao, Y., Wu, L., (2009). Second life: A new window for e-learning. *Proc. of IEEE 9^{th} Int. Conf. on Hybrid Intelligent Systems (HIS '09)* 3, 191–194, Washington, DC, USA,

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.*

Resilient Overlay Network for Real-Time Interactive Multimedia Sessions in Corporate Networks

Pelayo Nuño, Juan C. Granda, Julio Molleda, Rubén Usamentiaga, Daniel F. García

Department of Computer Engineering, University of Oviedo, Campus de Viesques, 33204 Gijón, Spain

Abstract

Real-time interactive multimedia communications are becoming increasingly useful for education, business, e-commerce and e-government, providing an enriched user experience in teleconferencing, e-meetings, distance training and product demonstrations. Large corporations are usually located at several sites, so real-time multipoint sessions within corporations are especially difficult. IP multicast is available or feasible within each site of an organization. Thus, corporate networks can be considered as various multicast-capable networks interconnected through a wide area network without multicast connectivity. This paper proposes a resilient self-managed overlay network to support real-time multipoint interactive sessions within corporate networks. The proposed overlay takes advantage of the configuration of corporate networks to self-organize and provide an efficient media delivery service, making use of multicast communications wherever available. Various self-healing techniques are implemented allowing for the continuity of ongoing sessions in spite of network disruptions and entity failures. Extensive simulations have been carried out to assess the performance and resilience of the overlay facing several types of disruptions.

Keywords: Overlay multicast, self-organization, self-healing, resilience, Session Initiation Protocol

1. Introduction

Overlay networks are complex data distribution networks deployed over existing networks [1] providing services unavailable in the latter. These services mainly include content distribution and peer-to-peer (P2P) data sharing, but they can also be used for real-time interactive communications provided that the latency of communications achieved is low [2]. All these services usually impose constraints on heterogeneity, scalability, availability and confidentiality of communications, making the deployment and management of overlay networks a true challenge. Thus, several issues might affect the resilience of the overlay, preventing users from communicating with each other.

Email address: `jcgranda@uniovi.es` (Juan C. Granda)

A Resilient Overlay Network (RON) should provide mechanisms to face unforeseen events, automatically recovering from failures in different entities of the overlay [3]. Disruption tolerance in RONs has been a challenge for the research community in recent years [4]. Many overlay networks propose self-organization and self-healing techniques, so communications can continue after failures. The main concern is to face failures in critical entities, for example a session controller, a rendezvous point, a bootstrap server or an authentication server. Traditionally, fault-tolerant hardware solutions, based on mirroring of critical entities and hot spare configurations, and software redundancy solutions have been used to avoid this issue. An overlay implementing a combined solution based on hardware and self-healing software techniques achieves a higher resilience level. Nevertheless, there are occasions where the former cannot be applicable. In those cases, self-healing techniques achieve a reasonable level of resilience allowing for ongoing sessions to continue although with limited functionality.

Real-time interactive communications are opening opportunities for business, e-commerce, e-government and education. Companies can benefit from interactive teleconferencing, e-meetings, product demonstrations and distance training [5], while universities may use real-time communications for developing innovative synchronous e-learning activities. The networks of these organizations are usually organized into multicast islands, one for each organization site, connected through virtual private network channels over the Internet [6]. This design makes the deployment of low-latency full-mesh overlay topologies feasible [7].

Disruptions may reduce the resilience of the overlay preventing users from participating in real-time sessions. This leads to a waste of useful time of users, which implies high costs for the organization as real-time interactive sessions (e-meetings, teleconferences, etc.) are usually scheduled during working time. The displacement of users to specially equipped rooms to participate in these sessions further increases the costs of service outages.

This paper introduces a resilient overlay network specially suited for real-time interactive multimedia communications in corporate networks. Various self-healing and self-optimization software techniques based on open standards are proposed to face the issues affecting the resilience of the overlay. These techniques improve the performance and resilience of the overlay. Real-time multimedia sessions (hereafter referred to as sessions) can continue with reduced functionality when a disruption occurs, and the proposed resilient techniques restore the session to a fully functional and managed state as soon as the disruption disappears.

The remainder of the paper is organized as follows. Background about resilient overlay networks is presented in Section 2. Section 3 briefly describes the architecture of the overlay network. Resilience concerns of the overlay architecture are detailed in Section 4. Section 5 details various software techniques constituting a resilience scheme to improve the overall resilience of the overlay. The resilience scheme is evaluated in Section 6 and the results exposed and discussed in Section 7. Finally, Section 8 contains the concluding remarks and outlines future work.

2. Background

Overlay networks are composed of many related entities providing services such as data distribution, session management and identification, authentication, etc. A survey of overlay networks for group communication services is presented in [8]. The authors classify overlay networks in several classes, although three classes are usually considered for developing time-constraint group communication services: automatic overlay multicast (a.k.a. Application Layer Multicast, ALM), P2P and unicast/multicast reflectors. Data delivery between endpoints and the management of the overlay usually relies on the endpoints themselves in ALM and P2P systems. On the other hand, a reflector is a proxy that forwards incoming multimedia traffic to one or more entities, which may be endpoints or other reflectors. Thus, it is possible to deploy an overlay network based on reflectors, avoiding routing and management tasks at the endpoints.

An overlay network can be deployed using several topologies to interconnect its members (endpoints for ALM and P2P, and reflectors for reflector-based overlays). The most common are trees and meshes. Two types of tree topologies are possible: shortest path trees and minimum spanning trees, focusing on minimizing latency and node degree respectively. Meshes are graph topologies where members can be interconnected in different ways. All members are connected to a central member in a star topology. In contrast, every member in the overlay is connected to each other in a full-mesh topology. A member of a ring topology is connected exclusively to two other members. A sun topology is similar to a star topology, but with a ring of members as the backbone of the topology rather than a single member. Finally, hybrid topologies are possible by combining basic topologies.

The efficiency of the data distribution depends largely on the topology of the overlay network. A low-node-degree topology such as a ring implies low data replication. However, the number of hops that data must traverse between senders and receivers is higher than in high-node-degree topologies (full-mesh or star), resulting in higher communication latencies. The main drawback of the latter is that their scalability may be limited due to data replication.

A RON is able to detect and recover from issues affecting its resilience within a short period of time [4]. RONs are mainly based on self-healing, self-organization and self-optimization mechanisms detecting network disruptions, recovering from them and dynamically discovering optimal network paths among members [9]. Measures such as packet loss rate, delay, jitter, round-trip time or application throughput are parameters frequently used when applying these mechanisms. Techniques based on idle times and the absence of refresh messages are also implemented [10].

A survey of the techniques used to provide resilience to overlay networks is presented in [11], classifying them in three approaches: cross-link, in-tree and multiple-tree redundancy. These approaches use path diversity to ensure that data reaches all the members of the overlay, which leads to a non-topological dependent data communication within the overlay. A resilient member is achieved with the help of several adjacent members and backup paths between them in cross-link and in-tree redundant techniques. Cross-link redundancy connects random members [12, 13, 14, 15], while in-tree redundancy organizes members in cluster-based trees and builds alternative paths within each cluster with members of other clusters [16, 17, 18]. In contrast, in multiple-tree redundancy several overlapped data distribution trees are built [19, 20, 21, 22].

Resilient techniques based on path diversity allow for building resilient overlays. However, these techniques do not suffice when failures occur in a critical entity, for instance a session controller. In that case, resilience can be achieved by using service replication [23]. There are two main approaches to perform replication: software redundancy techniques and fault-tolerant hardware solutions. Some active session members are used as backup of critical entities in software redundancy techniques. These members store the same information as the critical entities, and are able to play the roles of the latter in case of failure [24, 25]. In contrast, fault-tolerant hardware solutions are based on mirroring critical entities by standby members. Nevertheless, both approaches have some drawbacks. Software redundancy is not suitable for some real-time scenarios as it may severely penalize the performance of a backup member when taking the role of the critical entity. On the other hand, fault-tolerant hardware implies additional costs, especially when there are several critical entities in the overlay.

The proposed overlay uses the Session Initiation Protocol (SIP) to establish and re-establish sessions with a central entity when failures occur. To mitigate failures in this critical entity a resilience scheme is proposed. This includes various self-healing techniques allowing the central entity to recover the control of ongoing sessions after a failure. SIP has been previously used as part of robust and reliable P2P voice over IP (VoIP) services [26, 27] and multimedia communication overlays [28, 29]. However, to the best of the authors' knowledge, this work represents the first overlay network relying on standard SIP-based software mechanisms to achieve resilience. The resilience scheme controls the member level coping with failures at the endpoints, and the session level coping with risks derived from failures on the session controller. Although neither role redundancy nor mirroring support is used, the resilience of the overlay can be further improved by combining the proposed scheme with these two alternatives.

3. Overlay Network Architecture

The overlay network [30] is composed of three planes as illustrated in Figure 1. First, the relay mesh, represented by dashed lines, is responsible for forwarding multimedia data to participants. Multimedia data is delivered using the Real-time Transport Protocol (RTP), so several RTP relays are deployed composing a full-mesh overlay. Each relay works as a multicast/unicast reflector of RTP traffic. Second, the signaling plane allows users to participate in sessions and facilitates the negotiation of the multimedia configuration. This plane is represented by thin dotted lines in the figure. Finally, a mesh control plane, represented by solid lines, is used to reorganize the relay mesh according to the joining and leaving of participants and failures in the overlay in order to keep the real-time data delivery efficient. These planes are managed by a central entity called Rendezvous Point (RP). The RP regularly gathers link states from the relays through the mesh control plane and makes decisions about the organization of the relay mesh. Furthermore, the RP also manages the joining and leaving of participants through the signaling plane.

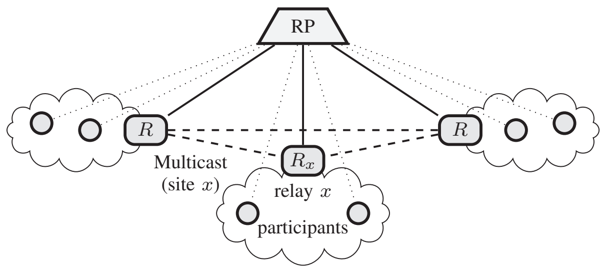


Figure 1: Architecture of the overlay

3.1. RTP Relay Mesh

Multimedia data must be delivered in real-time to enable interactive communications, so RTP is used to deliver audio and video. Separate RTP sessions are used to transmit each media type. An RTP session is not geographically or network scoped and must involve all the participants so they can send and receive data to and from the rest.

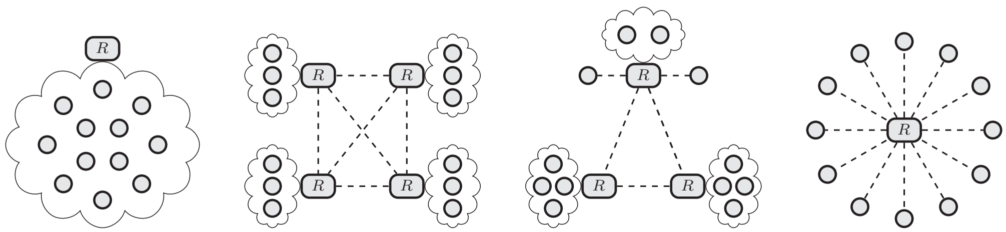
Ideally, IP multicast is available in the underlying network and each RTP session is associated to an IP multicast group, so participants must join two multicast groups (one for audio and another one for video) in order to receive all the multimedia data. However, the availability of IP multicast in corporate networks is limited. The corporate network can be considered as a set of IP multicast islands interconnected through a non-multicast-capable network.

An RTP relay (rectangle) is deployed in each IP multicast island and all the relays are connected in a full-mesh topology, as shown in Fig. 2. This topology minimizes end-to-end latency, as data from a participant (circle) must traverse a maximum of two relay hops. Furthermore, the use of a full-mesh topology makes the distribution tree trivial.

Every participant joins a session from a multicast island where a relay is located, so the participant may be associated with the relay. Thus, participants can communicate with the others by sending and receiving multicast streams. The relay forwards multicast streams from participants within the multicast island to peer relays in the relay mesh using unicast. It also forwards streams coming from peer relays to the participants using IP multicast. Thus, IP multicast is used where available, providing an efficient delivery service [7].

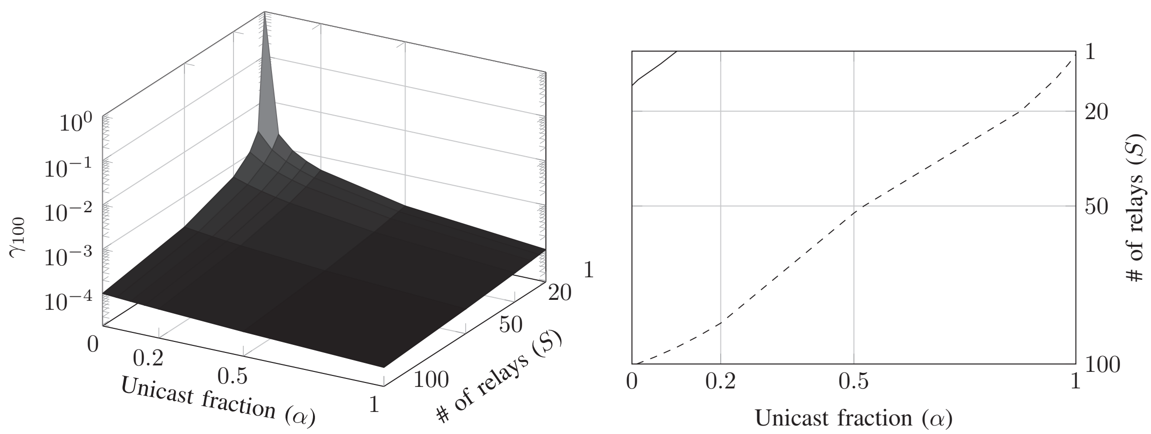
Occasionally, a relay may also forward traffic to participants without IP multicast connectivity. In this case, participants send and receive multimedia streams to and from the relay using unicast. Unicast delivery is required when the underlying network does not support IP multicast. This is also useful when the relay of the multicast island of a participant is down, so the participant can be redirected to another relay outside its multicast island.

The efficiency of the data distribution relies heavily on the availability of native multicast in the underlying network and the dispersion of participants. Ideally, all the participants are located in the same multicast island, so all multimedia traffic is



(a) $\gamma_{12} = 1$ (b) $\gamma_{12} = 0.027$ (c) $\gamma_{12} = 0.02$ (d) $\gamma_{12} = 0.0069$

Figure 2: Configurations of the relay mesh for 12 participants



(a) (b)

Figure 3: Value of γ_{100}

delivered without packet replication. As the number of multicast islands involved in the session increases, the relays have to forward traffic to more multicast islands, which increases packet replication. Redirected participants further decrease the efficiency of the data distribution. The lowest efficiency is obtained with a configuration of the mesh where all the participants are redirected to a single relay and traffic is delivered using unicast (a star topology). The number of multimedia streams flowing through the overlay depends on the number of multicast islands and the percentage of unicast participants. Assuming that a participant generates a single data stream, the number of streams flowing through the non-multicast-capable network of the relay mesh can be computed:

$$\alpha n^2 + (S - 1)n \tag{1}$$

where $S \geq 1$ is the number of relays (multicast islands), n is the total number of participants in the overlay and α is the fraction of participants using unicast delivery.

Next, a measure γ_n can be defined based on expression (1) to estimate the efficiency of a relay mesh configuration for n participants. The higher the value of γ_n, the lower the number of data streams flowing through the non-multicast-capable inter-network. This measure varies in the range $(0, 1]$:

$$\gamma_n = \frac{1}{\alpha n^2 + (S-1)n + 1} \tag{2}$$

Figure 2 shows the γ_n value for several configurations of the relay mesh with $n = 12$. Those mesh configurations with many relays in the mesh result in higher packet replication and bandwidth consumption. A value of $\gamma_n = 1$ refers to the most efficient configuration of the relay mesh, representing a relay mesh with only one multicast site. On the other hand, a γ_n value close to 0 refers to a low efficient configuration where most of the participants use unicast delivery. Figure 3 depicts the surface and contour of γ_{100}, *i.e.* a mesh with 100 participants, as a function of the number of sites and the fraction of participants using unicast delivery. As can be seen in the figure, the value of γ_{100} rapidly decreases when increasing the number of relays in the mesh or the fraction of unicast participants.

Although the γ_n value is a good estimation of the efficiency of the data delivery, it does not consider any imbalance in the resources of the relay mesh. Thus, two relay mesh configurations with the same values for S, n and α (with $\alpha > 0$) are assigned the same value for γ_n regardless which relays serve the unicast participants.

3.2. Signaling Plane

SIP is used for the management of sessions. The RP plays the role of a SIP focus to implement a tightly coupled conferencing service, so every participant establishes a SIP dialog with the RP while the session is running.

Participants join a session using the SIP handshake process, which makes the overlay network inter-operable with most SIP applications and devices, such as VoIP phones. The configuration of the multimedia session is described using the Session Description Protocol (SDP). Furthermore, it is possible to provide standard authentication mechanisms. The RP also implements a SIP registrar server, so participants can register to upload their current locations. Thus, it is possible to implement dial-out scenarios for joining a session, where the RP invites a registered user to participate in the session.

SIP also defines a mechanism to re-negotiate the configuration of the multimedia session. This may be used to develop self-healing techniques by changing the multimedia configuration of a participant affected by a network or relay failure.

3.3. Mesh Control Plane

The third plane is the mesh control plane. It is established between the RP and all the relays during sessions. The RP uses TCP connections with the relays to notify changes in the topology of the relay mesh. These changes are mainly due to inclusions and exclusions of relays from the mesh as participants join and leave or when network failures occur.

Relays must contact the RP as soon as possible to register with it, so the RP has an updated list of all the relays. It is not necessary that all the relays participate in the ongoing session. In fact, a relay will not be part of the relay mesh if there is no participant

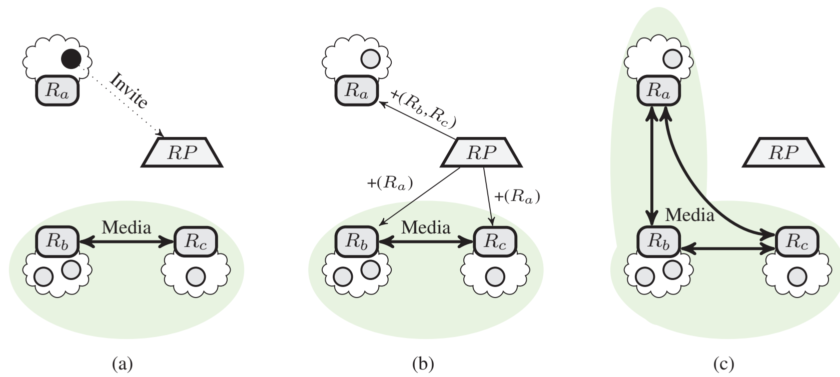


Figure 4: Inclusion of a relay in the relay mesh

in its multicast island. When a session begins, the RP disseminates the organization of the relay mesh to all the relays involved in the session. As a consequence, a relay may be in three states: unregistered, when it is not registered with the RP; registered, when it has registered with the RP; and active, when it is part of the relay mesh of the ongoing session and forwards multimedia data between the relay mesh and its participants.

3.4. Relay Mesh Self-Optimization

The overlay network includes a self-optimization technique that ensures a minimum number of streams interchanged among corporate sites, maximizing the value of γ_n. As previously mentioned, the RP maintains a list of registered relays, associating them with the identifier of the multicast island in which they are located. Since the relay mesh only involves active relays, the two basic processes involved in the optimization of the relay mesh are the inclusion and the exclusion of relays.

Figure 4 illustrates the inclusion of a relay in the relay mesh. The relay is included in the mesh when it becomes active. Figure 4(a) shows a configuration with three relays registered with the RP. There are three participants in the session: two located at the multicast island of relay R_b, and one participant at the multicast island of relay R_c (relay R_a is inactive). In this case, the configuration of the relay mesh is $\gamma_3 = 0.25$. A new participant at multicast island a establishes a SIP dialog with the RP to join the ongoing session. Therefore, as shown in Fig. 4(b), relay R_a becomes active and all the relays must be informed of the new mesh organization. The relay mesh is established between relays R_a, R_b and R_c. The RP sends an inclusion message to notify relays R_b and R_c that a new relay has been included in the relay mesh. When relays R_b and R_c receive the inclusion message, they add the peer relay included in the message to their forwarding lists. The RP also sends an inclusion message to relay R_a with information about all the active relays in the mesh. The inclusion messages contain information such as the IP address and a range of ports of one or more relays. This information is used by relays to forward traffic to their peer relays in the mesh. The RP generates inclusion messages according to the information provided by relays during the registration process. Fig. 4(c) shows the relay mesh deployed between multicast islands a, b and c after processing the inclusion messages. The final relay mesh organization is $\gamma_4 = 0.11$.

On the other hand, a relay is excluded from the relay mesh when it becomes inactive. The rest of the relays must be instructed to stop forwarding data to the inactive relay. If the participant at multicast island c in Fig. 4(c) leaves the session, the RP sends an exclusion message to relay R_c, so the relay processes the exclusion message and disconnects from the relay mesh. In addition, a removal message is sent by the RP to all the active relays (R_a and R_b) to notify that relay R_c has become inactive.

4. Resilience Concerns

Overlay networks must face various constraints resulting in complex systems with multiple points of failure where fault-tolerant hardware solutions are difficult to apply. Many distributed elements operate in the delivery of multimedia data and the admission of participants. A fault in a single element of the overlay may prevent the rest from participating in the session. Therefore, RONs should include self-healing techniques to increase resilience [31].

Many issues affecting the availability, reliability and security of an overlay network may arise causing disruptions. The failures originating disruptions can be characterized according to four criteria: domain, detectability, consistency and consequences [32]. Two failure domains can be considered: content failures and timing failures. The former refer to failures in which delivered data differs from that expected, while the latter refer to failures in which data arrival or data duration vary from that expected. Detectability addresses the signaling of failures, and distinguishes between failures that are detected and signaled and those that are not. Consistent failures are perceived identically across the overlay, while the perception of inconsistent failures may vary. The consequences of a failure may be minor or severe, affecting the availability, security, confidentiality or integrity of the overlay. Thus, consequences can be classified as fully-disruptive (FD), semi-disruptive (SD) and non-disruptive (ND) [33]. These consequences can be further classified as operational and functional consequences. Operational consequences are those involving the overlay network as a whole entity, leading to an overlay network disruption. On the other hand, functional consequences are those affecting only the quality of service.

Possible entity failures in the proposed overlay network and their consequences are summarized in Table 1. Firstly, not always does a failure in the RP or a network disruption affecting the RP imply fully-disruptive functional consequences. The RP is not responsible for delivering data, so the failure does not prevent participants from interchanging data. However, the self-management of the overlay is disabled, which avoids the triggering of the self-optimization technique when participants leave the session or the self-healing technique when other failures occur. In fact, in the best case RP failures lead to a semi-disruptive operational state, since resources are wasted due to relays with no participants not being pruned from the relay mesh. As a result, data streams are unnecessarily forwarded to these relays. In the worst case, an RP failure implies that the availability of the overlay decreases. New participants cannot establish

Table 1: Entity failures and consequences

Entity	Operational: Best case	Operational: Worst case	Functional: Best case	Functional: Worst case
Rendezvouz Point	SD	FD	ND	FD
RTP Relay	FD	FD	FD	FD
Participant	ND	SD	ND	FD

SIP dialogs with the RP, so they cannot join the session. Therefore, an RP failure in an incompletely established session prevents participants from taking part in the session leading to a fully-disruptive operational/functional state. The RP becomes a critical point of failure that must be considered when enhancing the resilience of the overlay network.

Secondly, a failure in one or several relays leads to a fully-disruptive operational/functional state. Although a relay failure only disconnects a section of the overlay network from the session, the session remains unavailable for the participants of the corporate site where the relay is located until the relay is recovered. Furthermore, network resources are wasted during such intervals since the rest of the relays in the mesh still forward data to the failed relay.

Finally, participant failures lead to non-disruptive operational states. However, when participants play different roles in the session, for example a presenter or moderator and many attendees, the failure of the presenter is more disruptive than the failure of one of the attendees. The presenter is temporarily disconnected from the session while the attendees remain joined in the worst case. Thus, this particular state can be considered non-disruptive at an operational level, but fully-disruptive at a functional level. In contrast, failures in any other participants cause a minimal functional perturbation. A failure makes the participant leave the session ungracefully without notifying the RP, which might lead to a waste of network resources implying a semi-disruptive operational state. This occurs when the last participant in a corporate site may have left the session ungracefully, since the rest of the relays keep forwarding data to the corporate site of the absent participant.

5. Resilience Scheme

Several software techniques are proposed to cope with the aforementioned disruptions and provide resilience to the overlay network. First, a technique for detecting failing entities by the RP is described. Next, a self-healing technique that allows the RP to gain control of an ongoing session after recovering from a critical failure is presented. A refined version of the algorithm for selecting the relay to which participants are redirected when their original relay fails is also formulated. Finally, a technique for recovering the control of an ongoing session after a failure in the RP is described.

5.1. *Failing Entities Detection*

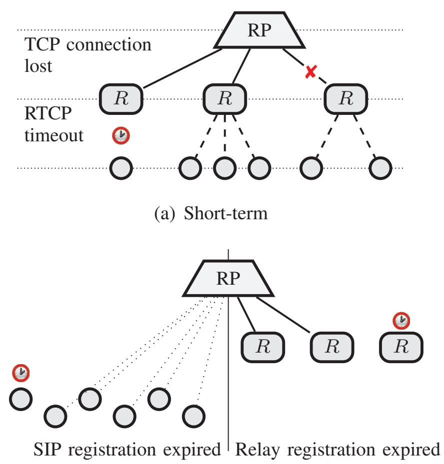
The efficient organization of the overlay network may be compromised due to failures of its entities. Firstly, a failure of the RP leads to a session in which new participants cannot join and the leaving of participants does not trigger the self-optimization mechanism. Secondly, a failure of a relay should cause the redirection of all of its associated participants. Finally, a failure of the last remaining participant of a multicast island implies a waste of resources, as data traffic is forwarded to a multicast island with no active participants. Therefore, it is critical to detect failures of these entities that may cause a reorganization of the relay mesh or loss of control over the session. Since the RP is responsible for the reorganization of the relay mesh, it must be able to detect failing entities, including itself. This is especially complex in the case of participants, as the communications between the RP and participants are not persistent. The RP and the participants interchange SIP messages over UDP, which is connectionless oriented, so the RP is not aware of participants leaving ungracefully. To solve this problem, a twofold mechanism for detecting failing entities is proposed: a short-term and a long-term detection mechanism.

The short-term detection mechanism tries to detect failing entities as quickly as possible building a three-level detection tree as illustrated in Fig. 5(a). The RP is the root of the tree, the relays are connected to the root, and the participants are the leaves of the tree. Each entity of the tree is responsible for detecting the failure of its descendants and notifying this fact to its ancestor. Thus, the relays notify the RP when one or more of its associated participants fail or leave the session ungracefully. This information is used by the RP to reorganize the relay mesh. A relay detects that a participant is not active by snooping RTCP traffic, which is the control traffic associated with RTP. When a relay receives no traffic from a participant, it considers that the participant has failed or abandoned the session. On the other hand, the RP establishes a TCP connection with all the active relays of the session, so the RP detects a failing relay by using the TCP connection as a keep-alive mechanism.

A long-term detection mechanism (Fig. 5(b)) is also proposed for discovering failures not reported by the short-term detection mechanism. It is based on the registration processes that must be carried out by relays and participants. Relays must register with the RP when they start running, while participants can be forced to perform a SIP registration with the RP acting as a SIP registrar. Both registrations expire, so relays and participants must refresh registrations periodically. In this case, the RP detects any failing entity directly. If the registration timer expires, the entity is considered as failed by the RP.

5.2. *Self-healing Technique*

The overlay network includes a cross-link redundancy self-healing technique to increase the overall resilience. Whenever a relay goes down, the RP redirects the participants of the relay to other active relays in the relay mesh. In this way, the participants still send and receive traffic to and from ongoing sessions. The self-healing technique allows for the continuity of the service in spite of relay failures. This technique collaborates with the self-optimization technique previously described to provide a resilient and efficient data delivery service.



(a) Short-term

(b) Long-term

Figure 5: Failing entities detection trees

Whenever the TCP connection between an active relay and the RP is lost, the RP sends SIP re-INVITE messages to the participants of the relay to change the configuration of the data delivery for those participants. These SIP messages redirect traffic from the participants to other relays using unicast. When the relay is up again and registers with the RP, the reassignment process is undone. The RP maintains a list of the changes made in the relay mesh, so it is able to undo any of them. In this case the self-optimization technique is applied to redirect participants to their original relay.

The redirection process has some drawbacks. Redirected participants must use unicast delivery to communicate with the relays, so an increase of network resource consumption occurs. Moreover, the algorithm used to select the relay to which participants are redirected is not optimal since only the number of participants currently associated with relays is taken into account [30]. The algorithm does not consider the conditions of the underlying network nor the resources available in each relay.

5.3. Refined Self-healing Technique

All the possible mesh configurations after triggering the self-healing technique receive the same γ_n regardless of the relays to which the participants are redirected. The main objective when redirecting participants is to balance the workload among the relays of the mesh. A refined version of the self-healing technique is proposed to balance the workload supported by each relay after the redirection process. This workload is measured in terms of the number of data streams that a relay receives and forwards using unicast. The traffic that the relay receives and sends within the multicast island is not considered, as the streams delivered using multicast are not replicated and the local network of a corporate site has plenty of resources compared to those available in its link to the Internet. In the case that a participant generates one data stream, the number of unicast data streams sent and received by relay i (θ_i) can be computed using the number of relays (S) and the number of participants served by the relay (n_i) [7]. The number of participants of relay i can be further divided in participants in the multicast island (m_i) and participants outside this, served using unicast (u_i). The relay must forward data to and from unicast participants and the rest of the relays in the mesh, so the expression computing θ_i is the following:

$$\theta_i = (S + n - 2)u_i + (S - 2)m_i + n \tag{3}$$

Assuming that all the relays have the same amount of resources available, the standard deviation of θ can be used as an indication of the workload distribution over the relay mesh. Therefore, the application of the self-healing technique becomes a search problem where $\sigma(\theta)$ must be minimized. A participant is redirected to the relay for which the resulting mesh configuration minimizes:

$$\sigma(\theta) = \sqrt{\frac{1}{S-1}\sum_{i=1}^{S}\left(\theta_i - \overline{\theta}\right)^2} \tag{4}$$

The above expression is minimum when all the relays support the same workload. However, not all the relays are able to deal with the same workload, as the available resources may vary from one relay to another. A balancing parameter ρ_i is defined in the range $(0, 1]$ to estimate the capacity of relay i to forward data to and from the rest of the relay mesh and unicast participants. A value of ρ close to 0 means that the relay has a small amount of resources to receive and forward unicast data streams. Thus, a measure $\delta_i = \theta_i/\rho_i$ is defined for a relay, so the function to minimize becomes:

$$\sigma(\delta) = \sqrt{\frac{1}{S-1}\sum_{i=1}^{S}\left(\delta_i - \overline{\delta}\right)^2} \tag{5}$$

The RP redirects a participant to the relay that minimizes the value of $\sigma(\delta)$. Thus, the workload is balanced over the relay mesh according to the resources available in each relay. The parameter ρ_i is critical to estimate the resources available in relay i. This parameter can either be established statically as a function of the network bandwidth of the relay link or it can be dynamically computed as network conditions change. However, a dynamically computed ρ_i requires that the RP gathers additional information from relays, which results in more control messages. Furthermore, rapidly changing network conditions may lead to excessive relay mesh reconfigurations. For simplicity, the relay with the highest network bandwidth is assigned $\rho_i = 1$ and the rest of the relays are assigned a value proportional to the network bandwidth available.

5.4. Self-recovery of Ongoing Sessions

The RP assumes that a failure has occurred when execution is stopped or when all the TCP connections with the relays are unexpectedly closed. The former may require a restart of the RP, while the latter implies a network failure that might be automatically recovered by the underlying network. In both cases the control over an ongoing session must be recovered, since relays keep forwarding data from participants. An algorithm

```
procedure OngoingSessionRecovery(session)
    ActiveRelays(session) ← ∅
    for all participant ∈ Participants(session) do
        relay ← OriginalRelay(session, participant)
        if ActivateRelay(session, relay) then
            sdp ← CreateSDP(session, relay)
            Reinvite(participant, sdp)
            Associate(session, relay, participant)
            ActiveRelays(session) ← relay
        else
            ReDirect(session, relay, participant)
        end if
    end for
end procedure

procedure ReDirect(session, relay, participant)
    repeat
        newrelay ← RedirectTo(Relays(session), relay, participant)
    until ActivateRelay(session, newrelay)
    sdp ← CreateSDP(session, newrelay)
    Reinvite(participant, sdp)
    Associate(session, newrelay, participant)
    ActiveRelays(session) ← newrelay
    MarkAsRedirected(session, newrelay, participant)
end procedure
```

Figure 6: Ongoing session control recovery algorithm

for self-recovery from failures of the RP is described in Fig. 6, provided that the state information of the session is continuously saved to a persistent database.

The state of the session is updated whenever a participant joins or leaves, a relay or a participant registers with the RP, or the relay mesh organization changes. This includes data such as the IP addresses of relays and participants, and the relationship between participants and relays. Information about redirected participants is also accounted.

The recovery of the ongoing session involves several steps. First, the RP recovers the state of the session from the persistent database and marks all the relays as inactive. Second, the RP tries to reestablish the SIP dialog with each participant. This implies activating the relay of the corporate site of the participant. If the relay is activated successfully, the RP creates an SDP description for the session and initiates a SIP dialog with the participant using a SIP re-INVITE message. If the connection with the relay cannot be re-established, the self-healing technique is triggered to redirect the participant to another relay.

6. Experimentation

A preliminary implementation of the RON has been successfully used to support real synchronous e-training activities [5]. However, an ns-3 model has been developed to perform exhaustive tests in order to assess the resilience of the overlay. The model simulates the operation of the network of a geographically dispersed organization and the participants within a multimedia interactive session.

A participant waits before joining the session. The waiting time is simulated using an exponential random variable $Exp(\lambda_w)$. Then, the participant establishes a SIP dialog with the RP in order to join the session and negotiate media configuration, and closes the dialog when leaving.

Typically, audio and video are the most commonly used media types during e-meetings, product demonstrations, e-training, etc. Thus, a participant is simulated as an entity that generates audio and video streams that must be forwarded to the rest of the participants. The audio and video streams of a participant are activated regularly. A video stream is a simulated 160×120 VBR H.264 video stream at 10 frames per second. An audio stream is a simulated CBR iLBC audio stream with a packetization time of 20 ms. Both the interval between activations and the duration of the streams are assumed to be normal random variables. Thus, a participant generates an audio stream of duration $\mathcal{N}(\mu_{al}, \sigma_{al}^2)$ after an elapsed time $\mathcal{N}(\mu_a, \sigma_a^2)$, and generates a video stream of duration $\mathcal{N}(\mu_{vl}, \sigma_{vl}^2)$ after an elapsed time $\mathcal{N}(\mu_v, \sigma_v^2)$.

Depending on the type of session, various participant roles can be identified. A product demonstration is carried out by a presenter, while all the participants usually play the same role in e-meetings. The presenter differs from a regular participant in that the former uses the audio and video channels exhaustively and the latter only when intervening in the session. All the tests include a presenter, who is simulated by a participant using the audio and video channels continuously.

In real sessions, there should be a maximum number of participants using the audio and video channels simultaneously to avoid excessive network bandwidth consumption. The intelligibility of data streams also depends on this number. Many participants transmitting audio at the same time will generate unintelligible information because of the overlapping of the audio streams. On the other hand, although many video streams do not overlap like audio streams, only a few can be rendered on screen. Therefore, floor control policies must be defined, so each participant in an interactive session can use the audio and video channels when holding the audio and video floors respectively. The responsibility of granting and revoking floors may rely on the presenter or on all the participants. In the former case the presenter, acting as a moderator, decides which partipants use the data channels at any given time. In the latter case, the participants holding the floors should release them so other participants can activate their audio and video streams. Specifically, a maximum of 4 audio streams and 4 video streams are allowed simultaneously during the tests. The floor requests from participants are granted in a FIFO order.

The corporate network is simulated as a set of multicast islands, each representing a corporate site as shown in Fig. 7. IP multicast is available in the local networks of the sites, while a non-multicast-capable network interconnects all the sites. A relay is deployed at each site so the participants can join the session through the relay mesh. Although many configurations may be possible such as various redundant or balanced network links, in the simulated scenario a site is connected to the Internet through a single link with a bandwidth B_x and an average delay of 25 ms. This bandwidth depends on the capacity of the underlying physical network and other data transfers using some of the available resources. Therefore, it is defined as a threshold so as to avoid interfering with the regular communications of the corporation, so the RP can be informed about the available bandwidth at each site before the session begins.

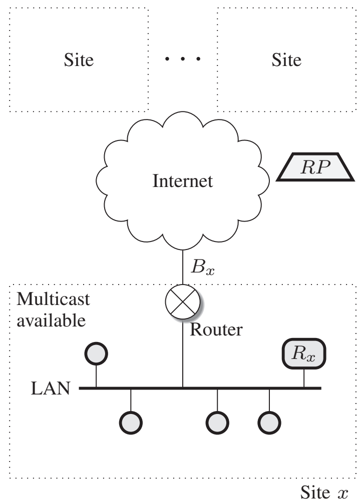


Figure 7: Simulated network environment

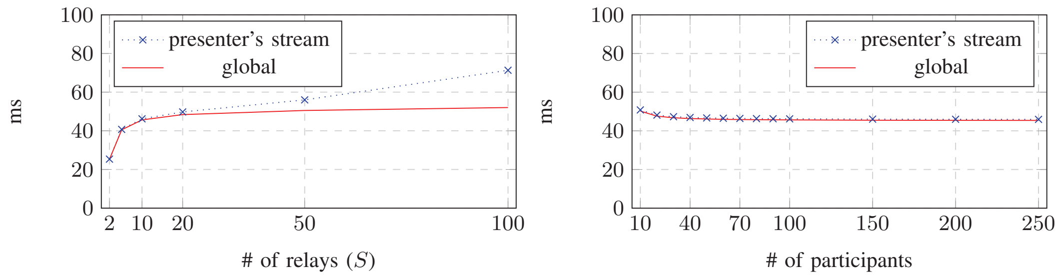
Table 2: Simulation settings

Waiting time to join (s)	λ_w	1/45
Session duration (s)	3600	
Audio activation interval (min)	(μ_a,σ_a)	(25,3.3)
Audio stream duration (s)	(μ_{al},σ_{al})	(15,2.6)
Video activation interval (min)	(μ_v,σ_v)	(40,8.2)
Video stream duration (s)	(μ_{vl},σ_{vl})	(30,9.5)

7. Results and Discussion

Simulations were conducted to assess the performance and the resilience of the overlay. Table 2 shows the parameters used during the simulations. Each test simulated a session of 1 hour and was repeated 7 times.

The user experience depends greatly on the latency of communications. A mouth-to-ear latency of 150 ms is barely noticeable, so it is commonly acceptable for interactive audio communications [34]. This latency includes the capturing and packetization latencies, the data delivery latency and the buffering time for playout at the receiver. Thus, the latency introduced by the overlay in the data delivery must be less than 150 ms to provide successful user experience. Video communications allow higher latencies, so this constraint only applies to audio streams. Figure 8 shows the average end-to-end latency observed in the audio packets from the presenter to the rest of the participants and the global average latency. All the sites have the same network bandwidth, that is $B_i = B_j : \forall i, j \in \{1, \ldots, S\}$, and the participants are balanced across the overlay so the relays serve the same number of participants. Figure 8(a) depicts the average latency observed at the receivers when increasing the dispersion of participants (number of relays) for a session with 100 participants. The figure proves that



(a) Session with 100 participants

(b) Session dispersed in 10 sites

Figure 8: Average end-to-end network latency

the global average latency grows asymptotically with the dispersion of participants. The workload of each relay increases since it has to forward data to more peer relays. However, since the use of the data channels is floor-controlled, only a few relays have active senders in their multicast islands and these vary throughout the session, so the increase in the average workload of relays is minimum as is the increase of the global latency. Conversely, the average latency observed in the audio stream coming from the presenter noticeably increases with the dispersion of participants. This is because the relay serving the presenter has to forward the data streams (audio and video) from the former to an increasing number of relays. Figure 8(b) shows the average latency as a function of the number of participants for a session with 10 sites. The increase in the number of participants per site and the use of floor-control policies to share the data channels makes more participants receive data from senders without passing through a relay (those co-located with the senders). Therefore, the latency slightly decreases with the number of participants. In any case, the latency keeps low throughout the session, so a successful user experience is achieved.

Although the average latency is low during the session, there are occasions when the latency may reach high values due to redirections of some participants and/or the presenter. Thus, an insight analysis of the dynamic end-to-end latency is carried out in Fig. 9. The figure shows the latency observed in the packets received from the presenter, located at site 1, at participant p, located at site 2, during a session with 100 participants balanced among 20 sites. The figure also shows the changes in the value of γ_{100} when some events occur. These events are described in Table 3. During the normal operation of the relay mesh, the latency of the packets is only influenced by the delay of the network path between sites 1 and 2, as seen in Fig. 10(a), since data from the presenter is directly forwarded by relay R_1 to relay R_2 and from the latter to the receiver. When relay R_1 goes down (event `d1`), the presenter is redirected to relay R_{10}, so the audio stream from the presenter must traverse two high-delay network paths as shown in Fig. 10(b). The latency further increases when relay R_2 also goes down (event `d2`). In this case, the receiver is redirected to relay R_{19} and the packets from the presenter must traverse one more high-delay path as shown in Fig. 10(c). This is the worst scenario, since the latency reaches the above-mentioned threshold of

Table 3: Events	
d1	Relay R_1 goes down
d2	Relay R_2 goes down
d3	Relay R_3 goes down
u3	Relay R_3 goes up again
u2	Relay R_2 goes up again
u1	Relay R_1 goes up again

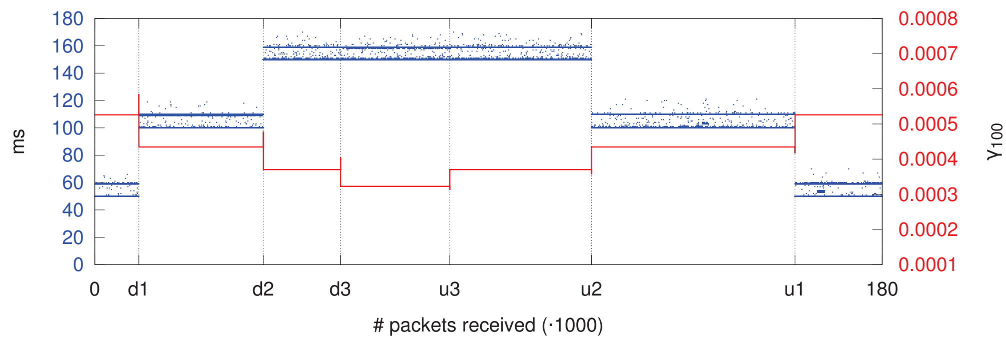


Figure 9: Dynamic end-to-end network latency

150 ms where the user experience may be affected. However, the impact on the user experience is minimum as the latency remains close to this threshold. A subsequent failure in another relay (event d3) and its recovery (event u3) have little influence on the observed latency at the receiver. When relays R_2 and R_1 are up again (events u2 and u1), the receiver and the presenter are redirected to their original relays so the latency lowers. The figure also shows that the value of γ_{100} decreases when the relays fail and increases when the relays are up again. The spikes in the value of γ_{100} when relays go down and up are due to participants being excluded from the session until they are redirected to other relays and relays being included in the relays mesh respectively.

Moreover, the time required for recovering the control of the session when a failure of the RP occurs is analyzed in Fig. 11. Figure 11(a) shows the time as a function of the number of relays for a session of 100 participants, and Fig. 11(b) depicts the time as a function of the number of participants for a relay mesh with 10 relays. Both plots consider a balanced scenario where all the relays have the same network resources and serve the same number of participants. As can be seen, the time to recover the

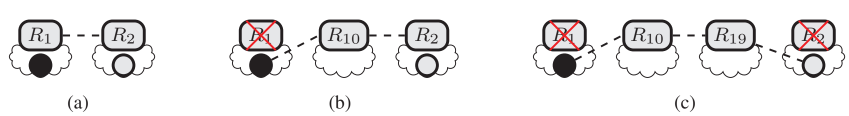
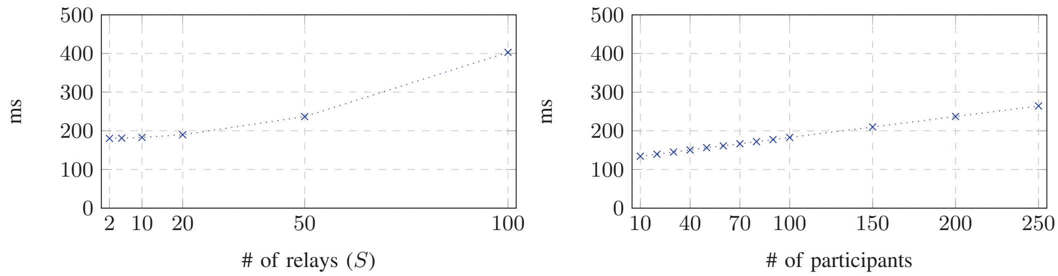


Figure 10: Details of the relay mesh: (a) before any event; (b) after event d1; (c) after event d2



(a) Session with 100 participants

(b) Session dispersed in 10 sites

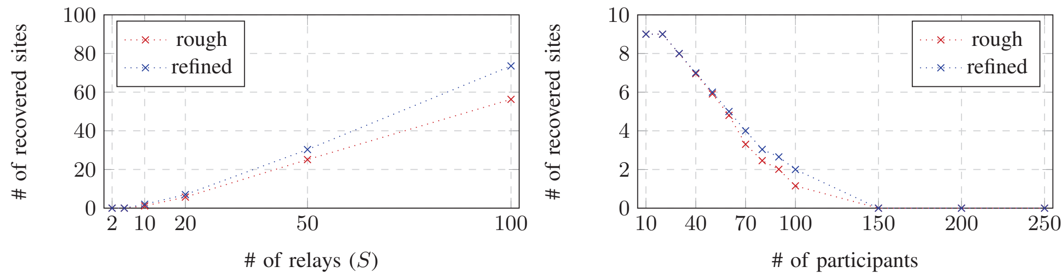
Figure 11: Time required for recovering the control of the session after an RP failure

session grows exponentially with the number of relays in the activity. This is because the size and number of control messages between the RP and the relays also grows exponentially. Each message must include all the peer relays for a relay. On the other hand, this time grows linearly with the number of participants in the session provided that the number of relays is constant.

The resilience of the overlay was also analyzed from various points of view. The self-healing technique used to redirect participants when a relay fails cannot cope with indefinite relay failures if relays do not recover. Redirected participants communicate using unicast, so the resources consumed significantly increase after a relay failure. Various tests were carried out to analyze the ability of the self-healing technique to redirect participants without compromising the whole overlay. The bandwidth available in each relay was simulated using a normal random variable B. Furthermore, all the participants sent a single audio stream continuously. Subsequent random relay failures were simulated during a session until one or more relays reached their maximum available bandwidth. These tests were repeated 1000 times.

Figure 12 shows the maximum number of sites (relays) that can be recovered by redirecting their participants when subsequent relay failures occur in a session with all the relays serving the same number of participants, but with an imbalance in the resources of relays. Specifically, the available bandwidth in Mbps of the links of the sites is $B = \mathcal{N}(20, 2^2)$. Figure 12(a) shows the maximum number of sites that can be recovered as a function of the number of relays in the mesh for a session with 100 participants, while Fig. 12(b) shows this as a function of the number of participants for a session with 10 relays. The refined version of the self-healing technique proves to be more efficient than the rough version for a high number of relays. The former adapts better to the imbalanced resources of relays by redirecting participants to those relays with more resources. The performance of both versions when varying the number of participants is similar. The difference in performance only appears for a high number of participants.

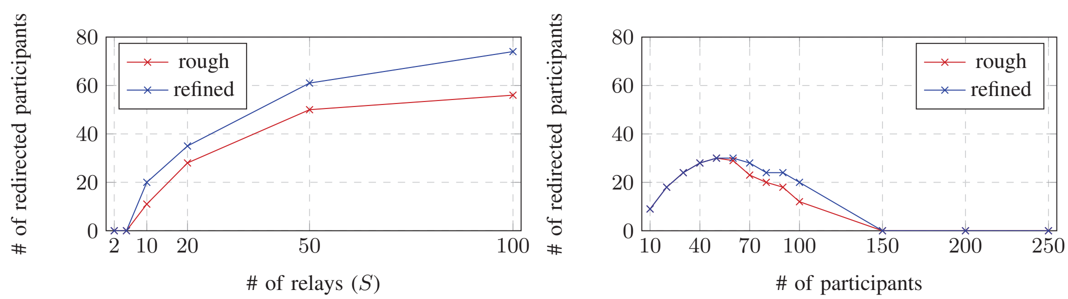
Figure 13 shows the maximum number of participants that can be redirected in the same conditions as for Fig. 12. Figure 13(a) shows the maximum number of participants that can be redirected when increasing the number of relays in the mesh for a



(a) Session with 100 participants

(b) Session dispersed in 10 sites

Figure 12: Maximum number of sites recovered after subsequent relay failures



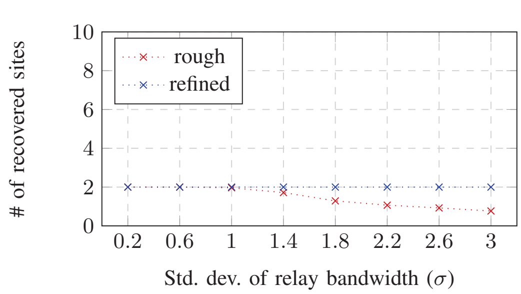
(a) Session with 100 participants

(b) Session dispersed in 10 sites

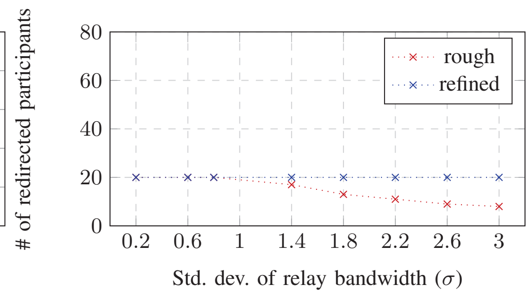
Figure 13: Maximum number of participants redirected after subsequent relay failures

session with 100 participants, while Fig. 13(b) shows this when the number of participants increases in a session with 10 relays. All the participants of a relay in a session with a low number of relays are impossible to redirect, since there is a high number of participants per relay. Therefore, the performance of the rough and refined techniques is the same for a low number of relays. However, when increasing the dispersion of participants (a higher number of relays) the differences between the rough and refined techniques can be identified. Besides, the refined technique performs better than the rough technique for a number of participants per relay higher than 6.

The influence of the imbalance of resources of relays when applying the self-healing technique is analyzed in Fig. 14. Figure 14(a) depicts the maximum number of sites that can be recovered when increasing the imbalance of the resources of the relays, while Fig. 14(b) shows the maximum number of participants that can be redirected in the same scenario. A session with 100 participants dispersed in 10 sites is simulated in both cases. Again, the performance of the refined technique is higher, but this is only evident for highly imbalanced bandwidths of the site links.



(a) Session: 100 participants, 10 sites



(b) Session: 100 participants, 10 sites

Figure 14: Maximum number of participants redirected and sites recovered when resources are imbalanced

8. Conclusions

A resilient overlay network has been proposed for real-time interactive multimedia sessions taking advantage of the specific configuration of corporate networks. This overlay is organized in three planes controlled by the RP: the RTP relay mesh, the signalling plane and the mesh control plane. This configuration is especially suited to geographically dispersed organizations. Various self-healing techniques are included to provide a resilient multimedia data delivery service among participants.

The use of standard protocols such as SIP for session signaling and RTP for data transport makes the overlay inter-operable with other software and hardware conferencing solutions, and the extensibility of the overlay is guaranteed. Furthermore, the self-healing techniques implemented are based on these standards, which is an application beyond their original purposes.

The relay mesh used to transport multimedia data makes efficient use of the available network resources. IP multicast is used where available, so participants communicate in real time, using low network resources. The full-mesh topology used enables a low-latency delivery service that provides an excellent user experience even when failures occur.

Moreover, the self-healing techniques implemented in the overlay make the configuration of the relay mesh automatic and transparent to the user, as well as robust in case of relay failures. The overlay is also prepared to recover from failures of the RP, which is a critical point.

Future work will be focused on developing self-regulation techniques allowing for transcoding data to cope with participant and site heterogeneity.

Acknowledgment

This research has been partially funded by the Spanish National Plan of Research, Development and Innovation under the project TIN2011-24903. Financial support (grant:UNOV-10-BECDOC-S) given by the University of Oviedo-Banco de Santander is also acknowledged.

References

[1] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, J. Zahorjan, Detour: Informed internet routing and transport, IEEE Micro 19 (1999) 50–59. doi:10.1109/40.748796.

[2] W. Zhang, Q. Zheng, H. Li, F. Tian, An overlay multicast protocol for live streaming and delay-guaranteed interactive media, J. Netw. Comput. Appl. 35 (2012) 20–28. doi:10.1016/j.jnca.2011.02.013.

[3] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, P. Smith, Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines, Comput. Netw. 54 (2010) 1245–1265. doi:10.1016/j.comnet.2010.03.005.

[4] D. Andersen, H. Balakrishnan, F. Kaashoek, R. Morris, Resilient overlay networks, in: Proceedings of the 18th ACM Symposium on Operating System Principles, Chateau Lake Louise, Banff, Canada, 2001, pp. 131–145. doi:10.1145/502059.502048.

[5] J. C. Granda, P. Nuño, F. J. Suárez, M. A. Pérez, E-pSyLon: A synchronous e-learning platform for staff training in large corporations, Multimed. Tools Appl. 66 (2013) 431–463. doi:10.1007/s11042-012-1061-9.

[6] Y.-W. E. Sung, S. G. Rao, G. G. Xie, D. A. Maltz, Towards systematic design of enterprise networks, IEEE/ACM Trans. Netw. 19 (2011) 695–708. doi:10.1109/TNET.2010.2089640.

[7] J. C. Granda, D. F. García, P. Nuño, F. J. Suárez, An efficient networking technique for synchronous e-learning platforms in corporate environments, Comput. Commun. 33 (2010) 1752–1766. doi:10.1016/j.comcom.2010.02.020.

[8] A. El-Sayed, V. Roca, L. Mathy, A survey of proposals for an alternative group communication service, IEEE Netw. 17 (2003) 46–51. doi:10.1109/MNET.2003.1174177.

[9] J. Li, P. L. Reiher, G. J. Popek, Resilient self-organizing overlay networks for security update delivery, IEEE J. Sel. Areas Commun. 22 (2004) 189–202. doi:10.1109/JSAC.2003.818808.

[10] A. Bikfalvi, J. García-Reinoso, I. Vidal, F. Valera, A peer-to-peer IPTV service architecture for the IP multimedia subsystem, Int. J. Comm. Syst. 23 (2010) 780–801. doi:10.1002/dac.1063.

[11] S. Birrer, F. E. Bustamante, A comparison of resilient overlay multicast approaches, IEEE J. Sel. Areas Commun. 25 (2007) 1695–1705. doi:10.1109/JSAC.2007.071208.

[12] X. Zhang, J. Liu, B. Li, T.-S. P. Yum, CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming, in: Proceedings of IEEE INFOCOM, Miami , FL, USA, 2005, pp. 2102–2111. doi:10.1109/INFCOM.2005.1498486.

[13] J. Liang, K. Nahrstedt, DagStream: Locality aware and failure resilient peer-to-peer streaming, in: Proceedings of the 13th SPIE/ACM Multimedia Computing and Networking, San Jose, CA, USA, 2006, pp. 127–138. doi:10.1117/12.643342.

[14] J. Shamsi, M. Brockmeyer, QoSMap: Achieving quality and resilience through overlay construction, in: Proceedings of the 4th IARIA International Conference on Internet and Web Applications and Services, Venice/Mestre, Italy, 2009, pp. 58–67. doi:10.1109/ICIW.2009.17.

[15] Z. Li, G. Xie, K. Hwang, Z. Li, Churn-resilient protocol for massive data dissemination in P2P networks, IEEE Trans. Parallel Distrib. Syst. 22 (2011) 1342–1349. doi:10.1109/TPDS.2011.15.

[16] S. Birrer, F. E. Bustamante, Resilient peer-to-peer multicast without the cost, in: Proceedings of the 12th SPIE/ACM Multimedia Computing and Networking, San Jose, CA, USA, 2005, pp. 113–120. doi:10.1117/12.587336.

[17] R. Tian, Y. Xiong, Q. Zhang, B. Li, B. Zhao, X. Li, Hybrid overlay structure based on random walks, Lect. Notes in Comput. Sci. 3640 (2005) 152–162. doi:10.1007/11558989_14.

[18] N.-F. Huang, Y.-J. Tzang, H.-Y. Chang, C.-W. Ho, Enhancing P2P overlay network architecture for live multimedia streaming, Inf. Sci. 180 (2010) 3210–3231. doi:10.1016/j.ins.2010.05.005.

[19] M. Castro, P. Druschel, A. Kermarrec, A.-M.and Nandi, A. Rowstron, A. Singh, SplitStream: High-bandwidth multicast in cooperative environments, ACM Oper. Syst. Rev. 37 (2003) 298–313.

[20] V. Venkataraman, K. Yoshida, P. Francis, Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast, in: Proceedings of the 14th IEEE International Conference on Network Protocols, Santa Barbara, CA, USA, 2006, pp. 2–11. doi:10.1109/ICNP.2006.320193.

[21] E. Setton, P. Baccichet, B. Giro, Peer-to-Peer live multicast: A video perspective, Proc. of the IEEE 96 (2008) 25–38. doi:10.1109/JPROC.2007.909925.

[22] A. H. Payberah, J. Dowling, F. Rahimian, S. Haridi, GradienTv: Market-based P2P live media streaming on the gradient overlay, in: Proceedings of the 10th International IFIP Conference on Distributed Applications and Interoperable Systems, Amsterdam, The Netherlands, 2010, pp. 212–225. doi:10.1007/978-3-642-13645-0_16.

[23] J. Zhang, L. Liu, L. Ramaswamy, C. Pu, PeerCast: Churn-resilient end system multicast on heterogeneous overlay networks, J. Netw. Comput. Appl. 31 (2008) 821–850. doi:10.1016/j.jnca.2007.05.001.

[24] J. Li, S. Vuong, An efficient clustered architecture for P2P networks, in: Proceedings of the 18th International Conference on Advanced Information Networking and Applications, Fukuoka, Japan, 2004, pp. 278–283.

[25] H. Yin, C. Lin, Q. Zhang, Z. Chen, D. Wu, TrustStream: A secure and scalable architecture for large-scale internet media streaming, IEEE Trans. Circuits Syst. Video Technol. 18 (2008) 1692 –1702. doi:10.1109/TCSVT.2008.927000.

[26] J. Fiedler, T. Kupka, T. Magedanz, M. Kleist, Reliable VoIP services using a peer-to-peer intranet, in: Proceedings of the 8th IEEE International Symposium on Multimedia, San Diego, CA, USA, 2006, pp. 121–128. doi:10.1109/ISM.2006.132.

[27] L. Andel, J. Kuthan, D. Sisalem, Distributed media server architecture for SIP using IP anycast, in: Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications, Atlanta, GA, USA, 2009. doi:10.1145/1595637.1595644.

[28] A. Fessi, H. Niedermayer, H. Kinkelin, G. Carle, A cooperative SIP infrastructure for highly reliable telecommunication services, in: Proceedings of the 1st International Conference on Principles, Systems and Applications of IP Telecommunications, New York City, NY, USA, 2007, pp. 29–38. doi:10.1145/1326304.1326310.

[29] G. Marchetto, M. Manzillo, L. Torrero, L. Ciminiera, F. Risso, Robustness analysis of an unstructured overlay for media communication, IET Commun. 5 (2011) 409–417. doi:10.1049/iet-com.2010.0624.

[30] J. C. Granda, P. Nuño, F. J. Suárez, D. F. García, Autonomic platform for synchronous e-training in dispersed organizations, J. of Netw. and Syst. Manag.

[31] J. C. Granda, P. Nuño, D. F. García, F. J. Suárez, Towards resilient synchronous e-training platforms, in: Proceedings of the 10th IEEE International Symposium on Network Computer and Applications, Cambridge, MA, USA, 2011, pp. 251–256. doi:10.1109/NCA.2011.42.

[32] A. Avižienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. Depend. Secur. Comput. 1 (2004) 11–33. doi:10.1109/TDSC.2004.2.

[33] R. Elling, I. Pramanick, J. Mauro, W. Bryson, D. Tang, Analytical Reliability, Availability, and Serviceability Benchmarks, Willey, 2008, Ch. 2, pp. 23–33.

[34] C. Perkins, RTP: Audio and Video for the Internet, Addison-Wesley Professional, 2003.

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 %
- FAU[2]: A

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

[1] Fuente: Microsoft Academic.

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

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 incluída en el *Book Citation Index* del Web of Knowledge (Thomson Reuters, 2013)

- Ranking Editorial: 17ª 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 APPLICATIONS 44/90 Q2:T2
  - COMPUTER SCIENCE, SOFTWARE ENGINEERING 24/105 Q1:T1

Bibliografía

[1] Adobe. Descripción de la herramienta en:
http://www.adobe.com/es/products/adobeconnect.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. Descripción de la herramienta 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. Descripción de la herramienta 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. Descripción de la herramienta 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. Descripción de la herramienta 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. Descripción de la herramienta 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 AACE World Conference on Educational Multimedia, Hypermedia&Telecommunications (EdMEDIA '07)*, Vancouver, Canada, June 2007.

[78] Mikogo. Descripción de la herramienta 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. Garcia, 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 Neurologia*, 35(6):553–562, 2002.

[83] Netviewer. Descripción de la herramienta en: http://www.netviewer.com.

[84] NS-3. Descripción de la herramienta 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. GradienTv: 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. Descripción de la herramienta 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. Sølie. 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. Descripción de la herramienta en: http://vivu.tv/vivuweb/support/vuroom/.

[121] Vyew. Descripción de la herramienta 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. Descripción de la herramienta 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 Transactions on Circuits and Systems for Video Technology*, 18(12):1692–1702, 2008.

[131] Yugma. Descripción de la herramienta en: https://www.yugma.com/.

[132] Yuuguu. Descripción de la herramienta 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 9^{th} 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. I. 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.