Estudio del rendimiento de arquitecturas basadas en grupos para WAHSN

Autor: García Pineda, Miguel
Director: Lloret Mauri, Jaime

Resumen —

Existen muchos trabajos relacionados con las redes ad hoc y las redes de sensores donde se presentan nuevos protocolos que encaminamiento que aportan mejores características, otros trabajos donde se comparan para ver cual posee un mejor rendimiento ó incluso presentan nuevas aplicaciones basadas en este tipo de redes, pero este trabajo aporta otro punto de vista. ¿Por que no ver la red como un conjunto que se divide en grupos para aportar un mejor rendimiento a la red independientemente del protocolo de encaminamiento utilizado?. Para ello, en este trabajo, vamos a demostrar a través de simulaciones, que la agrupación de nodos en redes WAHSN (Wireless Ad Hoc & Sensor Networks) aporta mejoras a la red en general, disminuyendo el tráfico de encaminamiento, el retardo, el throughput, etc. Este estudio se ha realizado evaluando los protocolos estándar más utilizados (DSR [1], AODV [2] y OLSR [3]), así podemos observar cual de ellos aporta un mejor rendimiento. Finalmente, se propone una arquitectura de red basada en grupos optimizada para las redes WAHSN.

Abstract —

There are many works related with ad hoc networks and sensor networks where the authors present new routing protocols with better or enhanced features, others just compare the performance of them or present an application environment, but this work tries to give another point of view. Why don't we see the network as a whole and split it intro groups to give better performance to the network regardless of the used routing protocol?. First, we will demonstrate, through simulations, that grouping nodes in WAHSN (Wireless Ad Hoc & Sensor Networks) improves the whole network by diminishing the routing traffic, the delay, the throughput, etc. This study was conducted to assess the most used standard protocols (DSR [1], AODV [2] and OLSR [3]) that gives better performance to the whole network when there are groups of nodes. Finally, a group-based network architecture optimized for WAHSN is proposed.

Autor: García Pineda, Miguel email: migarpi@posgrado.upv.es
Director: Lloret Mauri, Jaime email: jlloret@dcom.upv.es
Fecha de entrega: 12-09-08

ÍNDICE

I. INTRODUCCIÓN.

I.1. *Introducción.*

Las redes ad hoc inalámbricas también conocidas como WAHN (Wireless Ad Hoc Networks) son simples redes en las que no existe un nodo central que se encarga de gestionar la red y donde el número de nodos y la topología de la red no son predeterminados. Por otra parte las redes de sensores inalámbricas (WSN, Wireless Sensor Network) es un tipo de WAHN compuesto de nodos que tienen la capacidad detectar fenómenos que están ocurriendo en sus alrededores. Existen diferencias entre WSN y WAHN [4]. En las WSNs suele haber un mayor número de nodos y estos están desplegados en proximidad a los fenómenos que se quieran estudiar; los nodos utilizan principalmente el broadcast como modelo de difusión de datos y la topología de red puede cambiar constantemente debido, por ejemplo, al hecho de que los nodos pueden tener una probabilidad alta de fallo (tienen una potencia limitada, una capacidad computacional limitada y poca memoria). Las redes inalámbricas móviles de sensores (MWSNs, Mobile WSN) son WSNs con sensores móviles, que son aleatoriamente desplegadas en una zona de interés para la detección algunos fenómenos. Estos sensores colaboran con los demás para formar una red con la capacidad de percibir fenómenos y al mismo tiempo hacer una recopilación de datos en punto llamado sumidero o estación base.

Una red ad hoc móvil (MANET, Mobile Ad hoc NETwork [5]) es una red autoconfigurable de nodos móviles conectados por un medio inalámbrico. Este tipo de redes poseen una topología arbitraría. La topología de una red inalámbrica pueden cambiar rápidamente y estos cambios suelen ser impredecibles. Independientemente del método de acceso al medio utilizado [6], en los últimos años se han desarrollado muchos protocolos de encaminamiento para este tipo de redes [7] [8]. La movilidad de los nodos, la falta de estabilidad en la topología, la falta de una organización preestablecida y la utilización de las comunicaciones inalámbricas son algunas de las razones para no utilizar los protocolos de encaminamiento desarrollado para redes fijas.

Dependiendo del tipo de información intercambiada por los nodos y de la frecuencia con lo que lo hacen, los protocolos de enrutamiento en redes ad hoc están divididos en tres tipos: proactivos, reactivos e híbridos. Los protocolos proactivos actualizan la tabla de encaminamiento de todos los nodos periódicamente, aunque no exista información que intercambiar. Cuando ocurre un cambio en la topología, la tabla de enrutamiento se actualiza y el protocolo de encaminamiento considera cual es el mejor camino para transmitir la información desde un origen hacia un destino. Esto se realiza gracias a un mensaje de control periódico que se intercambia por toda la red, con la desventaja de aportar un consumo extra de ancho de banda y de energía en los nodos. Un ejemplo de este tipo de protocolos es el OLSR [3]. Los protocolos reactivos sólo mantienen y/o crean rutas en sus tablas de encaminamiento cuando un nodo tiene que comunicarse con otro nodo de la red.

Con estos protocolos, cuando una comunicación comienza, como la ruta correcta es desconocida, primero se envía un mensaje de descubrimiento de ruta. Cuando se recibe la respuesta, la ruta se incluye en la tabla de encaminamiento y después se establece la comunicación. La principal desventaja de estos protocolos es la latencia que existe al comienzo de las comunicaciones (tiempo de descubrimiento de ruta) pero se disminuye el tráfico de control de la red y los recursos energéticos. Los protocolos DSR [1] y AODV [2] son dos ejemplos estándar de este tipo de protocolos. Finalmente, los protocolos híbridos son una combinación de los dos tipos anteriores, tomando las ventajas de cada uno. Estos protocolos dividen las redes ad hoc en diferentes zonas, y por consiguiente los nodos cercanos usan un encaminamiento proactivo mientras que los nodos alejados usan encaminamiento reactivo. Dentro de este grupo aún no existe ningún protocolo normalizado.

Los tipos de redes y protocolos citados no sólo funcionan para una determinada topología, sino que podrían aplicarse en diferentes arquitecturas como grids, redes basadas en clusters, redes basadas en grupos y muchas más.

Un problema fundamental en la planificación de cualquier tipo de red es el diseño de la comunicación entre los diferentes nodos. En esta fase se debe decidir cómo los nodos establecen las conexiones entre si, así como cómo los mensajes intercambiados. Las topologías pueden caracterizarse por varios parámetros tales como: el número de nodos en la red, el número de conexiones en la red y su ancho de banda, el tipo de nodos y el diámetro de la topología. Por otra parte, el diseño de la topología lógica necesita abordar algunas exigencias contradictorias, por un lado, minimizar el diámetro de red, reducir al mínimo el tiempo de convergencia, el coste de la infraestructura (número total de enlaces), los costes de mantenimiento (por ejemplo; el número de enlaces mantenidos por cada nodo) y el coste de administración y, por otro lado, maximizar la distribución de carga, la fiabilidad, la eficiencia, la tolerancia a fallos, el rendimiento del sistema, la escalabilidad, etc. Generalmente, si optimizamos cualquier requisito esto provoca aumentar el coste de otros. El diseño de la topología óptima para un determinado conjunto de limitaciones es un problema que presenta cierta dificultad. Durante años, el diseño de topologías ha recibido un significativo interés en muchas áreas. A fin de proporcionar tiempo real en diversas infraestructuras, fiabilidad, disponibilidad, redes eficientes y servicios de distribución de contenidos que soporte una determinada QoS, por ello debemos ser conscientes que es necesario disponer de una adecuada topología de red para ofrecer los servicios de manera correcta [9] [10].

Aunque la topología física de la red define cómo están conectados físicamente los nodos de una red y el diseño físico de los mismos, la topología lógica define cómo los nodos de la red deben comunicarse (es decir, la manera en que los datos circulan por la red, con respecto a la interconexión física entre los dispositivos). Por tanto, es posible que nodos cercanos en la red lógica sean dispositivos muy alejados en la red física. Esto puede provocar el uso inadecuado de los recursos de red, y puede degradar la entrega de datos y el rendimiento significativamente.

I.2. *Objetivos.*

En este trabajo simularemos diferentes protocolos MANET, utilizando uno de los simuladores de redes más reconocido internacionalmente, OPNET Modeler [11], con el fin de evaluar el rendimiento según ciertos parámetros (cantidad de tráfico en la red cuando es estable, cantidad de tráfico en la red ante un cambio en la topología, tiempo de convergencia, número de actualizaciones que realiza el sistema, paquetes enviados/recibidos correctamente, paquetes enviados/recibidos con errores, etc.) cuando las redes están basadas en grupos.

El objetivo principal de este trabajo es evaluar los protocolos de encaminamiento nombrados en el punto anterior cuando se utilizan sobre un sistema basado en grupos y observar el rendimiento que aportan, con respecto a su funcionamiento en un sistema convencional. Todo ello sobre redes WAHSN.

A raíz de este objetivo aparecen otros objetivos secundarios como son:

- Aprender el funcionamiento y las características más importantes de las redes ad hoc, ya sean redes MANET ó redes de sensores inalámbricas.
- Conocer los diferentes tipos de sistemas de red y estudiar en profundidad aquellos que estén basadas en grupos.
- Analizar cuales pueden ser las posibles aplicaciones reales donde se podrían aplicar arquitecturas basadas en grupos.
- Manejar el simulador de redes con él cual se realizarán las simulaciones correspondientes para obtener los resultados de esta tesina.
- Analizar qué protocolos poseen mejor respuesta según lo que estén desempeñando, según la función que realicen o según el parámetro que se estén analizando.
- Introducir una nueva arquitectura basada en grupos, que se ajuste de la mejor manera posible a las características propias de las redes WAHSN.

Este análisis pretende servir de base y de justificante para diseñar una nueva arquitectura para redes de sensores basada en grupos, que posea un mejor comportamiento que las ya existentes.

I.3. *Precedentes de la tesina.*

Existe una gran variedad de trabajos relacionados con el tema de las redes MANET [6], [12] y otros diferentes, sobre las redes WSN [4], [5]. En algunos de ellos sólo se intenta mejorar alguno de los protocolos para unas determinadas actividades o aplicaciones. En otros, simplemente se observa el rendimiento del protocolo a través de simulaciones donde sólo se compara el caudal y el retardo que sufre la red cuando se utiliza un protocolo u otro.

En estos momentos la actividad que más interés está causando entre los investigadores de nuestra universidad respecto al tema de las redes de sensores inalámbricas es el estudio y la creación de protocolos adecuados para las redes WSN. Existen algunos trabajos finales de carrera presentados en las diferentes escuelas de la UPV, que estudian muchas de las características propias de las redes de sensores inalámbricas.

En el proyecto titulado "Propagation Model For Ad-Hoc & Sensor Networks", realizado por José Sabater Alepuz [13]. Trata los diferentes modelos de propagación existentes para las redes Ad-hoc y las redes de sensores. En él se intenta, a raíz de los diferentes modelos de propagación existentes, desarrollar un modelo adecuado para este tipo de redes. Este proyecto fue dirigido por Antonio Arnau Vives y presentado el 28/11/2007 en la ETSIT.

Otro proyecto realizado en la ETSIT, el cual ya trata tanto el análisis como el diseño, es el realizado por Carlos Domingo Ortiz [14]. En este proyecto se analizan las redes de sensores desde una visión mucho más amplia. Es decir, la red no sólo se extiende en un área sino también puede hacerlo en un volumen. Para ello se estudian diferentes diseños de redes de sensores en 2D y 3D, donde se observan diferentes modelos de transmisión, aspectos de cobertura, conectividad entre nodos, consumo de energía, etc. todo ello mediante simulaciones. Este proyecto se titula "Análisis Y Diseño De Redes De Sensores Inalámbricos 2D y 3D" y fue dirigido por Carlos Palau Salvador y defendido el 8/2/2007.

Las redes MANET, como veremos en capítulos posteriores, son un tipo redes Ad-hoc más maduras y estudiadas que las redes WSN aunque en éstas los estudios avanzan a pasos agigantados. Por esa razón ya poseen protocolos de red estándar [15] que son utilizados por diferentes dispositivos móviles o fijos.

Respecto a las redes Ad-hoc y MANET también existen diferentes trabajos realizados en esta universidad. Uno de los más importantes es la tesis doctoral de Carlos Miguel Tavares de Araújo Cesariny Calafate titulada "Analysis and design of efficient techniques for video transmission in IEEE 802.11 wireless ad hoc networks" [16]. Dicha tesis fue dirigida por Pietro Manzoni y defendida en el año 2006. En ella se realizado un estudio exhaustivo sobre que protocolos a nivel físico, MAC y red son los más adecuados para la transmisión de video en redes MANET. Con esta tesis se ha podido extraer mucha información respecto a este tipo de comunicaciones.

También existen proyectos final de carrera que trata este tipo de redes. Por ejemplo, el trabajo realizado por Luis Girones Quesada titulado "A Routing Protocol For Manets" [17]. Trata los diferentes protocolos de encaminamiento MANET. Para ello, son analizados una gran cantidad de los protocolos propuestos en la actualidad. Este proyecto fue dirigido por Antonio Arnau Vives y presentado el 25/9/2007 en la ETSIT.

Otro proyecto final de carrera que trata el tema de redes de sensores y redes MANET es él que realicé yo mismo. El proyecto titulado “Análisis y comparativa del rendimiento de los protocolos de encaminamiento MANET en redes de sensores inalámbricas” [18] trata de demostrar que es posible el uso de protocolos estándar MANET sobre redes de sensores inalámbricas. En él se observa que algunos protocolos ofrecen un buen comportamiento a nivel de red para las redes WSN. Este proyecto fue dirigido por Jaime Lloret Mauri y presentado el 11/12/2007 en la ETSIT.

Como hemos podido comprobar ninguno de los trabajos nombrados hasta el momento trata las redes como la unión de varios grupos. En la UPV sólo existe un trabajo donde se propone el uso de redes basadas en grupos para mejorar el rendimiento de la red como conjunto. Se trata de la tesis doctoral realizada por Jaime Lloret Mauri, titulada "Arquitectura de interconexión de redes P2P parcialmente centralizadas" [19] y dirigida por Manuel Esteve Domingo. Esta propuesta se basa en un nuevo sistema jerárquico para interconectar nodos de diferentes tipos de redes parcialmente centralizadas siempre que todas ellas compartan el mismo tipo de recursos (ficheros, contenidos, servicios, etc.) y permitirá compartir datos, contenido y recursos entre las redes P2P conectadas al sistema de interconexión. Se basa en la utilización de la capa de aplicación que permite agrupar nodos en redes lógicas. Esta tesis fue defendida el 27/7/06 en la ETSIT.

Debido al poco uso de las arquitecturas basadas en grupos en redes ad hoc y de sensores, y convencidos que este tipo de arquitecturas pueden aportar grandes mejoras a las WAHSN. Hemos creído conveniente realizar un estudio sobre el rendimiento que pueden aportar los protocolos desarrollados cuando son aplicados sobre arquitecturas basadas en grupos, pero en este caso se evitará depender de un sistema jerárquico como el propuesto en la tesis doctoral presentada por Jaime Lloret Mauri. Así, una vez comprobado que las arquitecturas basadas en grupos son más adecuadas para WAHSN desarrollar una arquitectura desde el inicio.

I.4. *Estructura de la tesina.*

La memoria de esta tesina se estructura en seis grandes bloques y un anexo. En el siguiente bloque se trata todo lo relacionado con topologías de red. En este punto se describe brevemente que es una topología y se muestran los tipos de topologías más comunes. A continuación en el mismo punto se presentan lo que son las topologías basadas en grupos y algunos ejemplos donde se utilizan.

En el bloque tres se habla de los posibles entornos de aplicación donde se podrían introducir y utilizar las topologías basadas en grupos de redes ad hoc y redes de sensores.

Seguidamente en el punto cuarto se realiza un estudio exhaustivo de los protocolos MANET más utilizados, cuando estos están trabajando sobre nodos con bajas capacidades (sensores) y se analiza los beneficios e inconvenientes que aportan estos protocolos a la red en general cuando se están utilizando topologías basadas en grupos.

En el quinto punto se presenta un esbozo ó los primeros pasos de desarrollo de una arquitectura para redes WAHSN basada en topologías basadas en grupos. En este punto se realiza una descripción de la arquitectura propuesta y a continuación se presenta los principios del modelo matemático que definen esta arquitectura y la selección de vecinos.

Por último, el sexto punto, se explicarán las conclusiones a las que hemos llegado mediante las simulaciones. También en este capítulo se presentarán los problemas o inconvenientes que hemos tenido durante el trabajo y las soluciones que hemos optado, así como los posibles trabajos futuros que pueden nacer a raíz de este trabajo.

II. REDES BASADAS EN GRUPOS.

II.1. *Tipos de redes.*

La topología de red define cómo los nodos de esa red están físicamente o lógicamente conectados (por ejemplo; la capa física de los dispositivos en la red). Podemos diferenciar tres tipos de redes:

- Redes centralizadas: En estas redes no puede haber una relación directa entre los nodos, todos los mensajes circulan por la red a través de un mediador, conocido generalmente como nodo central. Este nodo solo actúa como un gateway para todos los nodos. Esta forma de comunicación entre nodos se utiliza en muchos tipos de redes [20].
- Redes descentralizadas: Cada nodo es capaz de conectarse directamente con todos los demás nodos, y se envían mensajes sin la necesidad de utilizar un nodo central. Todos los nodos tienen la misma responsabilidad y funcionalidad en la red. Ningún elemento es esencial para el funcionamiento del sistema. Un nodo en una red descentralizada puede desempeñar tres funciones: servidor, cliente y router. Muchos tipos de redes poseen topologías descentralizadas, tales como las redes P2P puras, las redes ad-hoc y redes de sensores, etc. Se han diseñado y desarrollado muchos algoritmos para redes descentralizadas [21], pero en todos ellos se realizan tres acciones básicas: a) la búsqueda activa de nodos, b) interrogación por recursos o servicios, y c) la transferencia del contenido.
- Redes parcialmente centralizadas (también conocido como redes híbridas, o redes multinivel). En estas redes, hay algunos nodos con mayores funciones que forman la columna vertebral de la red y son necesarios para ejecutar el sistema. Otros nodos con menores funcionalidades se denominan nodos hoja y son colocados en la parte inferior de la capa lógica, otros nodos con mayor importancia se llaman supernodos y son colocados en los niveles más altos de la capa lógica de la red. Cada supernodo o nodo hoja puede tener conexiones con nodos hoja o supernodos. Existe una jerarquía donde los nodos de la capa superior realizan tareas organización, control o recopilación de datos de los nodos de la capa inferior. Los nodos de la capa superior son utilizados para transmitir mensajes desde los nodos de capas inferiores. Este tipo de topologías se utilizan por diferentes tipos de redes como redes de satélites [22], redes inalámbricas [23] e incluso como modelos para los procesos de negocios [24].

II.1. *Características de las redes basadas en grupos.*

Supongamos que necesitamos dividir nuestra red en grupos o zonas debido a la aplicación de la WAHSN o por un propósito de escalabilidad, y además, que no importe qué tipo de protocolo de encaminamiento se esté utilizando dentro de cada grupo. Todas las arquitecturas mostradas no pueden solucionar ese problema eficientemente. En el caso de arquitecturas centralizado, el servidor tendrá muchas conexiones inalámbricas al mismo tiempo, por lo que necesitará muchos más recursos. También es un punto central de fallo y un cuello de botella. Por otra parte, en el caso de las arquitecturas completamente distribuidas, es muy difícil de controlar el sistema y se necesita un largo periodo de tiempo para procesar algunas tareas (debido principalmente al tiempo necesario para llegar hasta nodos más lejanos), disminuyendo así el rendimiento del sistema.

En este trabajo proponemos dividir una WAHN o WSAN (Wireless Sensors and Actor Netwoks) en varios grupos y cuando un nodo recibe los datos de su grupo, este propagará la información al resto de los nodos en su grupo.

Un grupo se define como un pequeño número de nodos independientes con operaciones complementarias que interactúan con el fin de compartir recursos o tiempo computacional, o adquirir contenido o datos y así producir resultados comunes. En una arquitectura inalámbrica basada en grupos, un grupo consta de un conjunto de nodos que están cerca unos de otros (en términos de la ubicación geográfica, de área de cobertura o de tiempo ida y vuelta (RTT)) y los grupos vecinos pueden estar conectados si un nodo de un grupo está cerca de un nodo de otro grupo. El principal objetivo en las redes inalámbricas basadas en grupos es el protocolo de red y la gestión del grupo. Para ello, es necesario el diseño eficiente de un algoritmo y/o protocolo para encontrar el grupo más cercano (o mejor) para unirse cuando aparece un nuevo nodo en la red. El rendimiento de la red depende en gran medida de la eficiencia del proceso de localizar el grupo más cercano y de la correcta interacción entre grupos vecinos.

Hemos de distinguir entre arquitecturas groupware y arquitecturas basadas en grupo. En las arquitecturas groupware todos los nodos colaboran hacia el funcionamiento correcto y el éxito del propósito de la red, mientras que arquitecturas basadas en grupo toda la red se divide en grupos y cada grupo puede realizar diferentes operaciones y/o tener diferentes protocolos de encaminamiento.

Algunos aspectos importantes que se deben tener en cuenta en una arquitectura inalámbrica basada en grupos, independientemente del protocolo que se esté ejecutando dentro del grupo, son:

- Cómo construir los grupos vecinos.
- El protocolo empleado para intercambiar mensajes entre grupos vecinos.

Podemos distinguir entre dos tipos de arquitecturas basadas en grupos: las arquitecturas basadas en grupos planas y las arquitecturas basados en grupos de capas. En las redes basadas en grupos planas todos los nodos poseen el mismo rol y sólo hay una capa. Sin embargo, en algunos trabajos hay un servidor de directorios o un punto de encuentro (RP, Rendezvous Point) para la coordinación de la distribución de contenidos. En las redes basadas en grupos de capas, los nodos pueden tener varias funciones (2 funciones al menos). Dependiendo del tipo de función que estén realizando pertenecen a una capa u otra. Todos los nodos de la misma capa tendrán la misma función. En estas arquitecturas habrá conexiones entre nodos desde la misma capa y entre nodos de otras capas, pero en ambos casos las capas deben ser adyacentes. Hemos incluido las arquitecturas jerárquicas dentro de este grupo de redes de capas, porque las jerarquías podrían considerarse como capas.

Existen diferencias entre ambas arquitecturas basadas en grupos. Aunque las redes basadas en grupos con capas crecen de una forma estructurada, organizada por las mismas capas, las redes basadas en grupos planas crecen sin seguir una estructurada y sin ningún tipo de organización. Por otro lado, en las redes de capas cualquier nodo puede saber exactamente donde está cada grupo y cómo llegar a él. En cambio, las redes planas basadas en grupos, debido a que los grupos se unen a la red según aparecen, cada vez que hay una conexión entre nodos de diferentes grupos, el mensaje debe viajar a través muchos grupos desconocidos durante su camino. Los retardos entre los grupos de topologías en capas podrían ser inferiores porque las conexiones entre grupos pueden establecerse teniendo en cuenta este parámetro. En las topologías basadas en grupos planas, las conexiones entre grupos se establecen por la posición del grupo, su situación geográfica o su aparición en la red. Las redes con capas implican cierta complejidad porque los nodos pueden tener diferentes funciones y tolerancia a fallos lo cual implica un buen diseño en cada capa. Las redes planas son más sencillas porque todos los nodos tienen el mismo papel. Cuando hablamos de escalabilidad, las redes basadas en grupos con capas deben agregar más capas a su topología lógica, mientras que las redes basadas en grupos planas pueden crecer sin ninguna limitación, sólo debemos tener en cuenta el número de saltos del mensaje. Las redes basadas en grupos proporcionan algunos beneficios para toda la red, tales como:

- Propagación eficiente de los datos a través de los grupos de la red, dando mayor flexibilidad, y menos retardos.
- Aumentará la disponibilidad del contenido, podría repetirse en otros grupos.
- Cualquiera podría obtener datos de cada grupo sólo utilizando un servicio.
- Tolerancia a fallos. Otros grupos podrían llevar a cabo tareas cuando exista el fracaso de un grupo.
- Escalabilidad. Un nuevo nodo puede sumarse a cualquier grupo y un nuevo grupo podrían añadirse fácilmente a la red.
- El comportamiento de la red podría ser tomado y evaluado en cualquier grupo.

Existen algunos trabajos en la literatura donde la red de nodos está dividida en grupos y se establecen conexiones entre los nodos de diferentes grupos, pero todos ellos se han desarrollado para resolver temas específicos ([25] [26] [27] y [28]), pero ninguno para redes MANET.

A. Wierzbicki et al. presentaron Rhubarb [25] en 2002. Rhubarb organiza los nodos en redes virtuales permitiendo conexiones a través de cortafuegos o routers que utilizan NAT (es su objetivo principal), y además permite enviar broadcasts entre éstos eficientemente. Los nodos pueden ser activos, si establecen conexiones, o pasivos, si no lo hacen. Este sistema tiene sólo un coordinador por grupo y los coordinadores se pueden agrupar en grupos de manera jerárquica. Los nodos en Rhubarb establecen conexiones TCP permanentes con el coordinador proxy (un nodo activo que está fuera de la red privada). Esta conexión se debe renovar cuando se rompe por un firewall o un router haciendo NAT. Si un nodo que está fuera de la red desea comunicarse con un nodo dentro de la red, debe enviar una petición de conexión al coordinador proxy que reenviará la petición al nodo que está dentro de la red. Rhubarb utiliza una jerarquía de grupos de 3 niveles y cada 100 nodos se genera un nuevo grupo. El problema principal es que la escalabilidad de esta arquitectura no es muy grande.

Z. Xiang et al. presentaron el artículo "a Peer-to-Peer Based Multimedia Distribution Service" [26] en 2004. Este artículo propone una topología lógica en la cual los hosts cercanos se auto-organizan en grupos de aplicación. Los hosts dentro del mismo grupo tienen condiciones de red similares y pueden colaborar fácilmente entre ellos para conseguir que exista calidad de servicio. Cuando un nodo en esta arquitectura desea comunicarse con un nodo de otro grupo, la información se encamina a través de varios grupos hasta que llegue al destino deseado.

Existen algunas arquitecturas jerárquicas donde los nodos están estructurados jerárquicamente y partes del árbol que forman esa arquitectura constituyen grupos, determinados ejemplos se pueden observar en las referencias [27] y [28]. En ciertos casos, algunos nodos tienen conexiones con otros nodos de otros grupos aunque estos estén en diferentes capas del árbol, pero en todos los casos, la información tiene que ser encaminada mediante la jerarquía lógica de la red.

También hay otras arquitecturas jerárquicas basadas en clusters [29]. En una arquitectura basada en clusters los nodos móviles están divididos en grupos virtuales. Cada grupo posee conexiones con sus grupos adyacentes. Todos los grupos deben cumplir las mismas reglas. Un grupo está formado por un nodo Cluster Head, otros nodos Cluster Gateways y otros Cluster Members ([30] [31]). El Cluster Head es el nodo padre del grupo, él es que gestiona y comprueba la situación de los enlaces en el grupo, y la información correcta de rutas hacia otros clusters. El resto de los nodos de un cluster son todos nodos hoja. En este tipo de red, los nodos Cluster Head tienen un control total del grupo y el tamaño del grupo es normalmente de 1 ó 2 saltos desde el Cluster Head. Los Cluster Gateways tienen enlaces con otros nodos de otros clusters y la información de ruta hacia los otros grupos. Por otro lado, un Cluster Member es un nodo sin enlaces entre nodos de otros clusters. Finalmente, queremos destacar que las redes basadas en clusters son un subconjunto de las redes basadas en grupos, porque cada cluster puede ser considerado como un grupo. Pero una red basada en grupos es capaz de tener cualquier tipo de topología dentro de un grupo, no sólo clusters. Sin embargo, ambos tipos de redes se han creado para resolver los problemas de la escalabilidad de las WAHSN.

También hemos encontrado en la literatura protocolos de encaminamiento basados en zonas. Se trata del protocolo Zone Routing Protocol (ZRP) [32] [33]. En este protocolo cada nodo de modo proactivo mantiene información de encaminamiento con un conjunto de vecinos local (zona de encaminamiento), mientras que de manera reactiva adquiere rutas a destinos más allá de la zona de encaminamiento. ZRP y nuestra propuesta tienen varias características comunes, por ejemplo, que podrían aplicarse sobre cualquier tipo de protocolo de enrutamiento, ellos escalan bien y la información se envía a los nodos frontera con el fin de alcanzar destinos fuera de sus zonas. La principal diferencia entre ellos es que en ZRP cada nodo mantiene una zona y los nodos en esa zona tienen diferentes nodos en su zona mientras que en nuestra propuesta todos los nodos que forman un grupo tienen los mismos nodos en su grupo.

Por otro lado, no hemos considerado los siguientes trabajos como sistemas basados en grupos como tal. El modelo de movilidad para redes ad hoc basado en comunidades presentado en [34], porque aunque la red está organizado en grupos, y los nodos pueden pasar de una categoría a otra, no hay ninguna conexiones entre los nodos frontera de los diferentes grupos. En la jerarquía Landmark presentada en [35] aunque existe un nodo con mayor rol que tiene conexiones con nodos de otros grupos, sus nodos hoja no. Otro ejemplo similar a este último es la arquitectura del protocolo enrutamiento BGP [36]. Finalmente, no se va a considerar el movimiento de grupos como en Landmark Routing Protocol (LANMAR [37]), donde el conjunto de nodos se mueven como un grupo, de tal modo que el grupo puede aumentar o disminuir de tamaño con el movimiento de los nodos de la red.

III. ENTORNOS DE APLICACIÓN.

III.1. *Posibles entornos de uso.*

Las redes basadas en grupos se podrían utilizar cuando se quiere configurar una red donde pueden aparecer grupos y unirse a la red en cualquier momento o cuando la red se tiene que dividir en zonas más pequeñas para soportar un gran número de nodos, es decir, en cualquier sistema donde los dispositivos estén agrupados y deban existir conexiones entre los grupos.

La siguiente lista muestra diferentes áreas de aplicación donde se podrían utilizar WAHSN basadas en grupos:

- Supongamos un empleo donde todos los recursos humanos necesitan ser divididos en grupos para lograr un propósito (un escuadrón de bomberos quiere extinguir un fuego). Ahora, vamos suponer que todas las personas que participan en esta actividad necesitan un dispositivo que tiene que estar conectado con otros dispositivos del mismo grupo para recibir información de los miembros dentro del grupo, y de los grupos cercanos para coordinar sus esfuerzos. Actualmente coordinación entre los grupos se realiza mediante una conexión inalámbrica a un centro de mando o utilizando las comunicaciones vía satélite. Pero, algunas veces, ninguna de esas soluciones puede utilizarse porque se necesita una línea de visión directa, porque existen demasiados muros y por lo tanto la señal no tiene suficiente potencia para alcanzar el destino.

- Para comunicaciones en campos de batalla, es especialmente útil para la comunicación entre escuadrones para colaborar cuando un objetivo es blanco de los detectores de posición.
- Los grupos también podrían ser establecidos según ubicaciones o desniveles geográficos. Esto ocurre especialmente en las zonas rurales y entornos agrícolas. En este tipo de medio ambiente podría ser útil una arquitectura basada en grupos para detectar plagas o incendios y así propagar una alarma a zonas próximas y realizar las tareas oportunas. Se proporcionaría una gestión y control de detección de incendios y plagas más eficiente y además se permitiría la escalabilidad.
- Monitorización de la salud [38]. Un paciente podría necesitar ser monitorizado en varios lugares mientras él realiza una actividad. Cada sala o lugar podría tener uno o varios grupos de sensores (e incluso cada grupo disponer de diferentes tipos de topología en el interior) y los grupos vecinos deberían comunicarse para guardar las muestras y datos de los pacientes.
- Podría ser utilizado en cualquier tipo de sistema en el que un evento o alarma está relacionado con lo que está sucediendo en una zona específica, pero condiciona a los eventos que están ocurriendo en zonas vecinas. Un ejemplo es un sistema basado en grupos para medir el impacto ambiental de un lugar. Podría ser mejor si las mediciones son tomadas de diferentes grupos de sensores, pero los grupos de sensores tiene que estar conectado con el fin de estimar la totalidad del impacto ambiental. Otro ejemplo donde se podrían utilizar este tipo de agrupaciones es en las redes submarinas de sensores UWAN (Underwater Acoustic Networks) [39]. Los sensores podrían agruparse de manera adecuada para detectar diferentes aspectos subacuáticos y realizar una comunicación entre grupos para alertar o informar de cualquier evento al resto de sensores de nuestra red y también al centro de control situado en un punto de la costa.
- Juegos virtuales basados es grupos. Hay muchos juegos donde los jugadores se agrupan virtualmente para realizar una tarea específica. Las interacciones entre los grupos en la realidad virtual debe darse por interacciones entre los jugadores de diferentes grupos para intercambiar sus conocimientos.

En el siguiente punto vamos a demostrar que las redes basadas en grupos aportan un mejor rendimiento que los sistemas corrientes de WAHSN. Además se verá el rendimiento de los tres protocolos MANET más comunes y se analizará cuál de ellos posee un mejor comportamiento cuando los nodos se estructuran en grupos.

IV. RENDIMIENTO DE LAS REDES WAHSN BASADAS EN GRUPOS.

IV.1. *Banco de pruebas.*

En primer lugar vamos a presentar el banco de pruebas utilizando en este trabajo para todos los protocolos analizados. Para cada protocolo hemos simulado cuatro escenarios: el primero con nodos fijos, el segundo con nodos móviles y con errores, el siguiente para nodos agrupados y el último para nodos móviles con errores agrupados. Para cada topología, hemos realizado simulaciones con 100 y 250 nodos para así poder observar la escalabilidad del sistema. Los resultados se han obtenido mediante el simulador OPNET Modeler [11].

Las topologías escogidas para ser simuladas no siguen ninguna estructura estándar, sino que hemos escogido una topología aleatoria para el conjunto de las simulaciones, que nos ha facilitado OPNET Modeler. El hecho de coger una topología aleatoria se debe a que como estamos trabajando con redes inalámbricas los nodos van cambiando de posición constantemente y por tanto la topología física no sigue ningún patrón a priori conocido. Además los nodos se mueven aleatoriamente desde que empieza la simulación hasta que termina. Por ello los datos obtenidos en los puntos posteriores no dependen ni de la topología inicial de los nodos ni del patrón de movimiento de los mismos ya que todo es aleatorio.

Para realizar la simulación, hemos provocado fallos en la red con sus respectivas recuperaciones. Esto se realiza para observar el comportamiento de la red, ya no sólo ante cambios en la topología física, sino también ante fallos en los nodos que forman dicha red. Estos eventos de fallos y recuperaciones, en este tipo de redes suceden habitualmente ya que es común que algún nodo falle, que un usuario desconecte su nodo, etc. Por ello, se debe de estudiar cómo funciona un protocolo de nivel de red ante estos eventos.

Para la topología de 100 nodos se han creado 6 grupos que cubren aproximadamente un área circular de 150 metros de radio, en cada grupo existen unos 16 o 17 nodos aproximadamente. El número de nodos de un grupo varía con el tiempo debido a la movilidad aleatoria que poseen los nodos, por tanto en un instante determinado pueden pertenecer a un grupo y en otro instante a otro. En la topología de 250 nodos ocurre lo mismo, en este caso el número de grupos es igual a 12, lo que supone que habrá unos 15 o 16 nodos por grupo, muy similar al escenario anterior. El número de nodos que puede haber en el grupo varía con el tiempo por la misma razón que en el caso anterior. Las áreas cubiertas en ambas topologías por los grupos son similares.

Los nodos de la topología creada tienen las características propias de un nodo ad hoc, es decir, un procesador a 40 MHz, una memoria de 512 KB, un canal radio de tasa máxima 1 Mbps, utilizando la frecuencia de trabajo de 2.4 GHz. Hemos decidido que los nodos tengan un radio de cobertura máximo de 50 metros, parámetro conservador ya que la mayoría de los nodos de redes ad hoc poseen mayor radio de cobertura. Pero hemos preferido tener menor potencia de transmisión para cada dispositivo ad hoc y así ampliar su tiempo de vida.

El tráfico inyectado en las simulaciones corresponde con el tráfico MANET generado por OPNET Modeler. Éste empieza a los cien segundos de dar comienzo la simulación, posee un régimen de llegadas que sigue una distribución de Poisson con un tiempo medio entre llegadas de 30 segundos. El tamaño del paquete sigue una distribución exponencial con media de 1024 bits. El tráfico inyectado posee una dirección destino aleatoria, para así obtener una simulación independiente de hacia donde vaya dirigido dicho trafico de datos. Nosotros hemos simulado ambos escenarios para los protocolos DSR, AODV y OLSR. Los resultados obtenidos se muestran en los siguientes apartados.

IV.2. *Retardo medio en la capa de aplicación.*

En la Fig. 1. y Fig. 2. vemos el retardo medio a nivel de aplicación que sufre la información transmitida. En la Fig. 1. observamos que las arquitecturas basadas en grupos sufren un retardo medio muy cercano a 0.005 segundos independientemente del número de nodos que existan en la red. En el caso de las topologías clásicas este retardo es igual a 0.02 segundos en el escenario de 100 nodos y de 0.03 segundos en él 250 nodos, una vez la red converge. En el caso de la topología de 100 nodos existe una mejora de un 75% que aumenta en un 83% en el caso de 250 nodos.

Si nos fijamos en la Fig. 2. (escenario con movilidad y errores) vemos que los retardos a nivel de aplicación son más elevados en las arquitecturas basadas en grupos hasta que la red converge. En este caso vemos que hasta el instante 1300 segundos las arquitecturas basadas en grupo presentan un peor comportamiento. A partir de ese punto el retardo disminuye, en este caso el factor de mejora es bajo (alrededor del 5%).

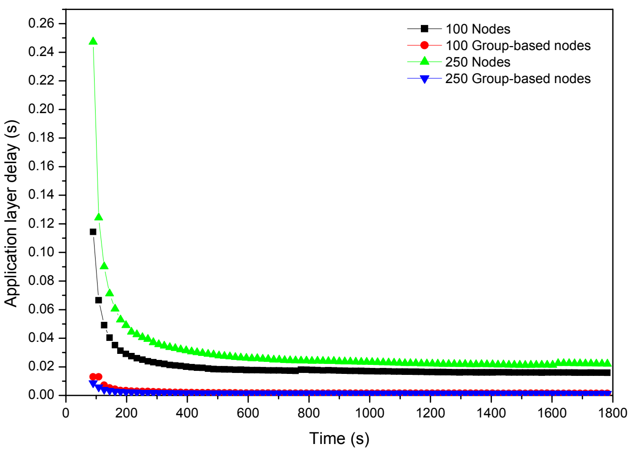


Fig. 1. Retardo medio a nivel de aplicación en topologías fijas utilizando el protocolo DSR.

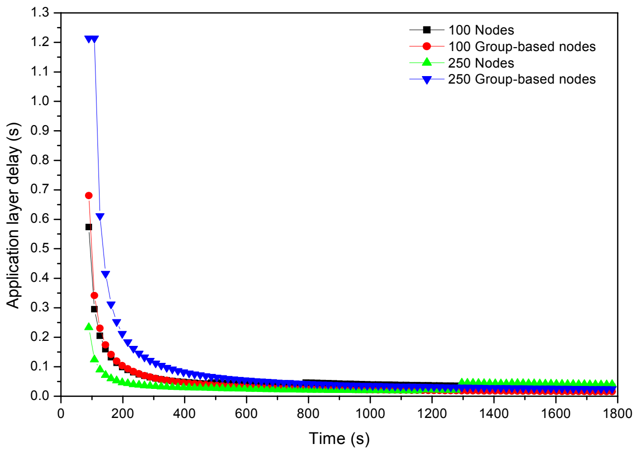


Fig. 2. Retardo medio a nivel de aplicación en topologías móviles y con errores utilizando el protocolo DSR.

El retardo medio a nivel de aplicación para el protocolo AODV se puede ver en las Fig. 3 y 4. Utilizando este protocolo en una topología fija (ver Fig. 3) se observa que dicho retardo no posee una gran dependencia del número de nodos. Para las topologías de 100 y 250 nodos tenemos un retardo superior a 0.5 segundos una vez ha convergido la red, mientras tanto existen picos que pueden llegar a los 2.5 segundos. En cambio para las topologías basadas en grupo este retardo es similar y se sitúa entorno a los 0.15 segundos. Las topologías basadas en grupos mejoran en un 70%.

Si nuestro escenario posee movilidad y posibles fallos obtenemos la simulación de la Fig. 4. En ella vemos que en el caso de 250 nodos tenemos un retardo de 1 segundo en la fase de régimen permanente, en esta misma fase la topología clásica de 100 nodos tiene un retardo medio aproximado de unos 0.75 segundos. Cuando estas mismas topologías están basadas en grupos el retardo disminuye por debajo de los 0.25 segundos en ambos casos. Aportando un mejora de un 67% en el peor de los casos.

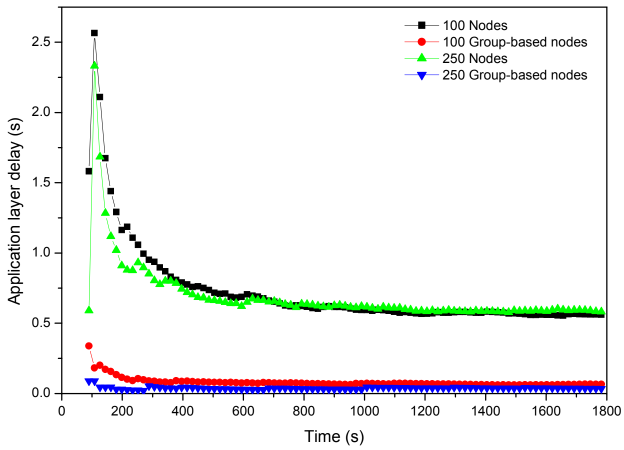


Fig. 3. Retardo medio a nivel de aplicación en topologías fijas utilizando el protocolo AODV.

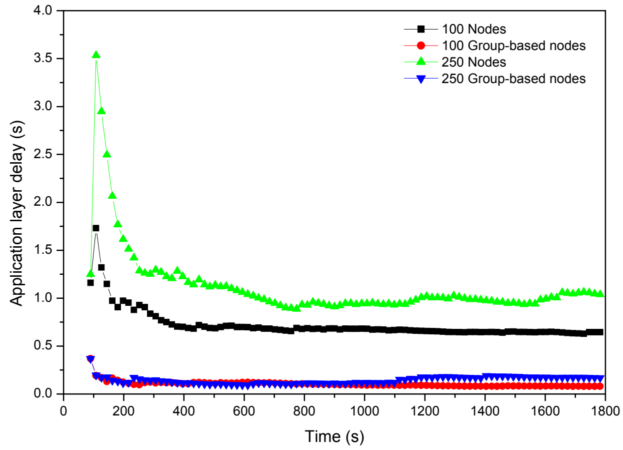


Fig. 4. Retardo medio a nivel de aplicación en topologías móviles y con errores utilizando el protocolo AODV.

En la Fig. 5. tenemos el retardo simulado a nivel de aplicación que sufren las topologías fijas. En el caso de 250 nodos obtenemos un retardo alrededor de los 0.015 segundos que pasa a valer 0.0035 segundos en el caso de la arquitectura de 250 nodos basada en grupo, aportando una mejora del 76%. En el caso de 100 nodos la mejora que aporta la arquitectura basada en grupos es inferior, como podemos ver la topología clásica tiene un retardo cercano a 0.005 segundos y cuando esta basada en grupos este retardo medio disminuye a los 0.002 segundos (60% de mejora).

En el caso que exista movilidad y fallos en los nodos (ver Fig. 6.) estos datos ya cambian. Se puede observar el caso de las topologías de 100 nodos, tenemos un retardo medio igual a 0.007 segundos una ver ha convergido la arquitectura clásica que disminuye a 0.0025 segundos en el caso del escenario basado en grupos (64% de mejora). En el caso de 250 nodos la mejora no es tan alta y muy similar a la obtenida en la simulación de la arquitectura de 100 nodos fijos, pasamos de 0.005 segundos a 0.002 segundos.

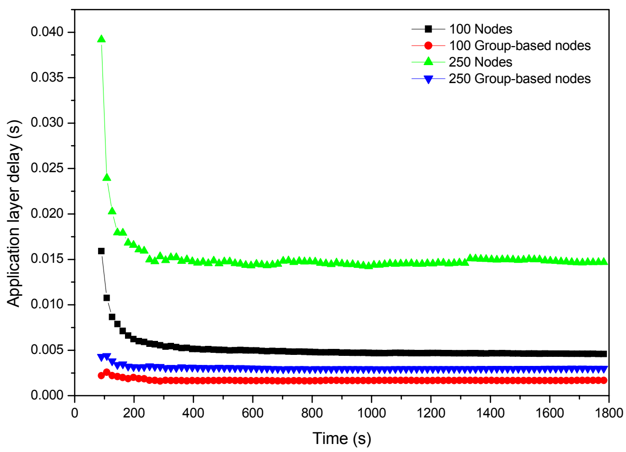


Fig. 5. Retardo medio a nivel de aplicación en topologías fijas utilizando el protocolo OLSR.

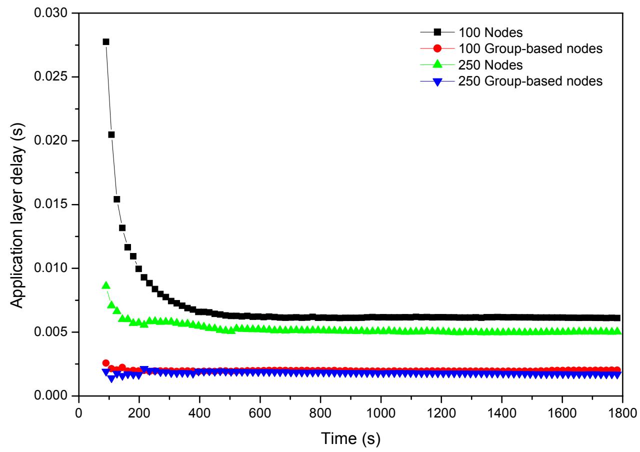


Fig. 6. Retardo medio a nivel de aplicación en topologías móviles y con errores utilizando el protocolo OLSR.

IV.3. *Tráfico de encaminamiento recibido.*

Seguidamente comparamos el tráfico de encaminamiento recibido utilizando el protocolo DSR (Fig. 7. y Fig. 8.). En el caso de la Fig. 7. observamos que el tráfico es bastante estable, esto se debe a las propias características de la red, ya que se trata de un topología fija y sin errores (ideal, no real). El tráfico recibido en la topología de 250 nodos esta en torno a los 500 Kbits/s en cambio con el mismo número de nodos pero basados en grupos vemos que este tráfico disminuye hasta los 200 Kbits/s (aportando una mejora del 60%). En torno a ese mismo valor esta el tráfico de encaminamiento enviado por los 100 nodos (250 Kbits/s), el cual sufre una mejora del 60% cuando tenemos los 100 nodos basados en grupos (100 Kbits/s).

En la Fig. 8. observamos que el comportamiento es muy similar. En este caso se puede observar que cuando existen errores en la topología de 250 nodos (intervalo 600-800 segundos y en torno a los 1200 segundos) el tráfico fluctúa mucho y es menos estable. Esta inestabilidad se puede observar que es mucho menor en las arquitecturas basadas en grupos.

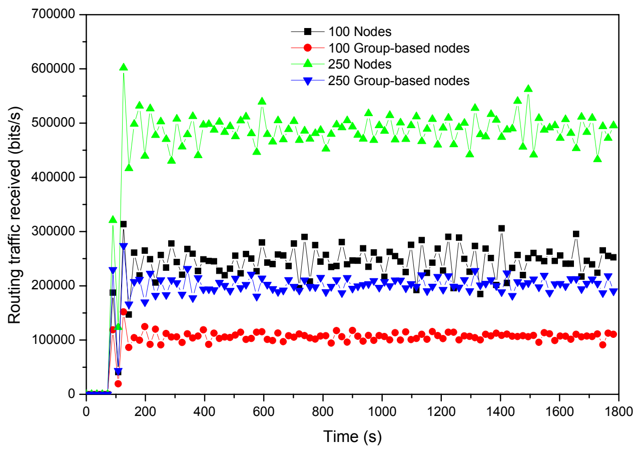


Fig. 7. Tráfico de encaminamiento recibido en topologías fijas utilizando el protocolo DSR.

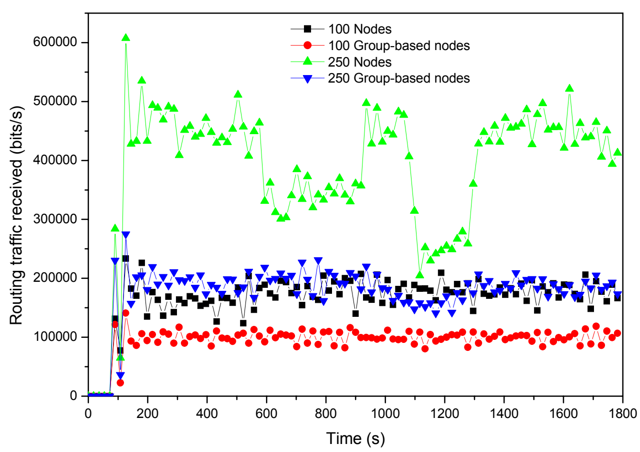


Fig. 8. Tráfico de encaminamiento recibido en topologías móviles y con errores utilizando el protocolo DSR.

El tráfico de encaminamiento AODV recibido para topologías fijas y móviles con errores se puede ver en la Fig. 9. y Fig. 10. respectivamente. Cabe destacar que al utilizar el protocolo AODV el efecto de inestabilidad visto en la Fig. 8. para el protocolo DSR desaparece. Podemos observar que el tráfico de encaminamiento es independiente de la movilidad de los nodos. Para la Fig. 9. tenemos un tráfico de encaminamiento de unos 460 Kbits/s, que para el caso de 250 nodos pasa a ser 260 Kbits/s cuando utilizados la arquitectura basada en grupo (43% de mejora). En el caso de 100 nodos pasamos de unos 230 Kbits/s a unos 140 Kbits/s, aportando así la topología basada en grupos una mejora del 39%.

Cuando existe movilidad y errores (ver Fig. 10.) la topología de 250 nodos pasa de 440 Kbits/s a los 250 Kbits/s en el escenario basado en grupos mejorando en un 43% la cantidad de tráfico de encaminamiento que circula por la red. Para la topología clásica de 100 nodos tenemos 200 Kbits/s y en la basada en grupos unos 135 Kbits/s (mejora del 32%).

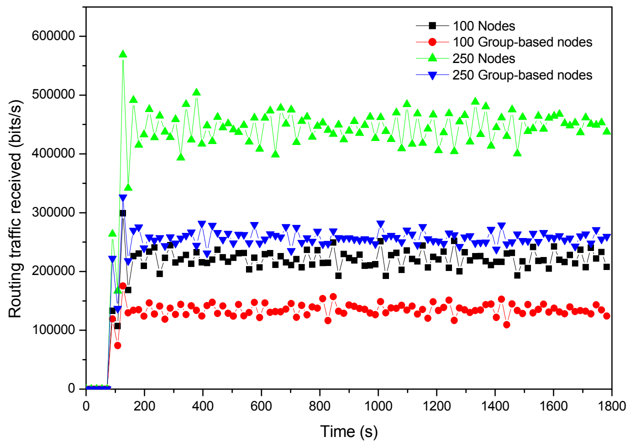


Fig. 9. Tráfico de encaminamiento recibido en topologías fijas utilizando el protocolo AODV.

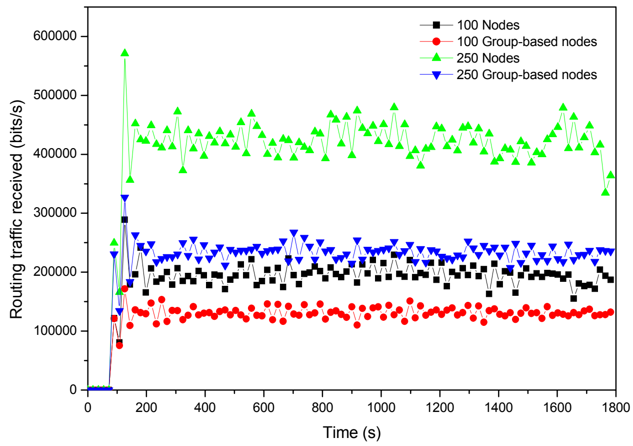


Fig. 10. Tráfico de encaminamiento recibido en topologías móviles y con errores utilizando el protocolo AODV.

Por último vamos a ver el comportamiento del protocolo OLSR a través del tráfico medio de encaminamiento enviado en las topologías fijas y móviles con errores (ver Fig. 11. y Fig. 12., respectivamente).

El tráfico de encaminamiento recibido en la red de 100 nodos fijos se sitúa alrededor de los 180 Kbits/s, cuando esta misma topología está basada en grupos dicho tráfico disminuye hasta los 70 Kbits/s, aportando una mejora de un 61%. En el caso del escenario de 250 nodos vemos que el tráfico medio de encaminamiento se sitúa sobre 300 Kbits/s, si este mismo escenario lo simulamos para una arquitectura basada en grupos el resultado es inferior al obtenido en el caso de 100 nodos fijos sin grupos (inferior a 150 Kbits/s) mejorando la carga de la red en un 50%, (ver Fig. 11).

Si analizamos el mismo parámetro cuando nuestra red posee movilidad y posibilidad de fallos debemos de fijarnos en la Fig. 12. El tráfico de encaminamiento es bastante sensible a los fallos que se producen en la red, por esa razón tanto en la traza de 100 nodos como en la de 250 nodos se puede apreciar diferentes fluctuaciones debidas a las propias características de la red simulada. Estas fluctuaciones se reducen cuando poseemos arquitecturas basadas en grupo. Respecto a la mejora que introducen este tipo de topologías basadas en grupo cuando existe movilidad y fallos es muy similar a la vista en topologías fijas. Existe una mejora superior al 61% en la arquitectura de

100 nodos y un mejora de un 50% en la de 250 nodos. Podemos prestar atención en como las topologías basadas en grupos suavizan las fluctuaciones de tráfico producidas por los fallos de los nodos.

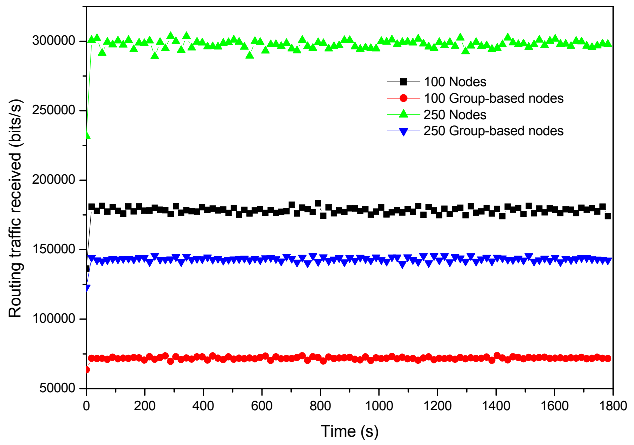


Fig. 11. Tráfico de encaminamiento recibido en topologías fijas utilizando el protocolo OLSR.

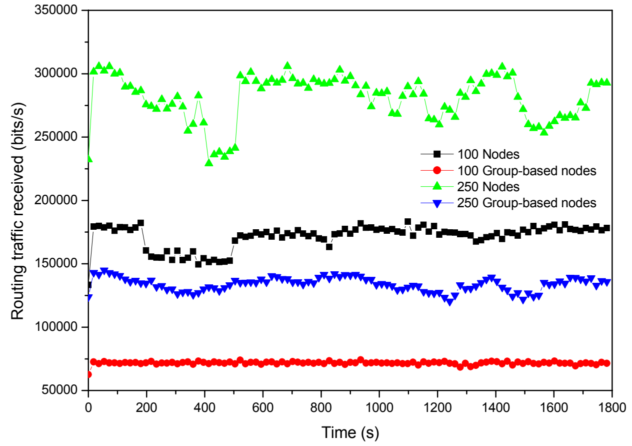


Fig. 12. Tráfico de encaminamiento recibido en topologías móviles y con errores utilizando el protocolo OLSR.

IV.4. *Retardo medio a nivel MAC.*

En la Fig. 13. y Fig. 14. vemos el retardo medio a nivel MAC que sufre la información transmitida. En la Fig. 13. observamos que las topologías basadas en grupos sufren un retardo medio de unos 0.00025 segundos independientemente del número de nodos que existan en la red. En el caso de las topologías corrientes este retardo es igual a 0.0011 segundos una vez la red converge. La diferencia entre ambos casos está sobre los 0.00085 segundos, disminuyendo por tanto el retardo medio a nivel MAC un 77% en ambos casos.

Si nos fijamos en la Fig. 14. los retardos son inferiores, esto es debido principalmente a la movilidad de la red. En este caso vemos que existen diferencias entre las arquitecturas de 100 y 250 nodos, cuando tenemos topologías basadas en grupos vemos que el retardo existente a nivel MAC es muy similar, en ambas topologías poseemos un retardo cercano 0.0001 segundos.

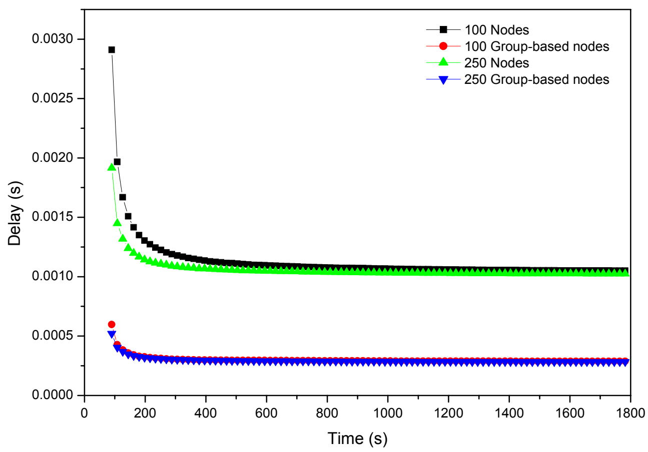


Fig. 13. Retardo medio en topologías fijas utilizando el protocolo DSR.

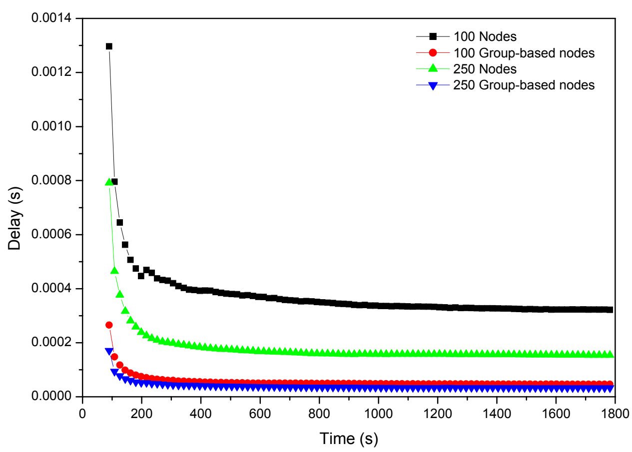


Fig. 14. Retardo medio en topologías móviles y con errores utilizando el protocolo DSR.

El retardo medio a nivel MAC para el protocolo AODV se ve en la Fig. 15 y en la Fig. 16. Con este protocolo se observa que dicho retardo no posee una gran dependencia del tipo de topología ni del número de nodos. Para las topologías de 100 y 250 nodos tenemos un retardo que se estabiliza en los 0.001 segundos, en cambio para las topologías basadas en grupo este retardo es igual a 0.0001 segundo. Las topologías basadas en grupos aportan una mejora de en un orden de magnitud.

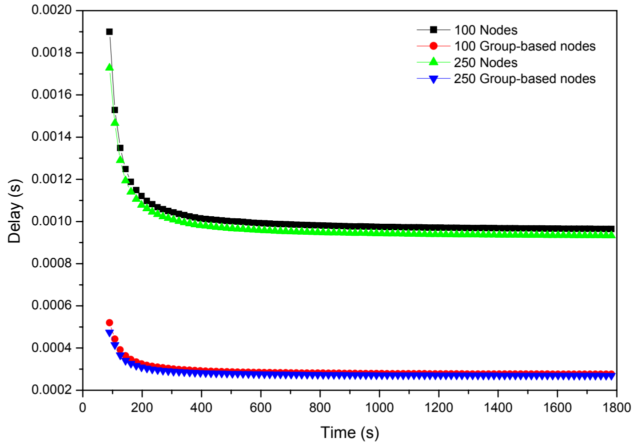


Fig. 15. Retardo medio en topologías fijas utilizando el protocolo AODV.

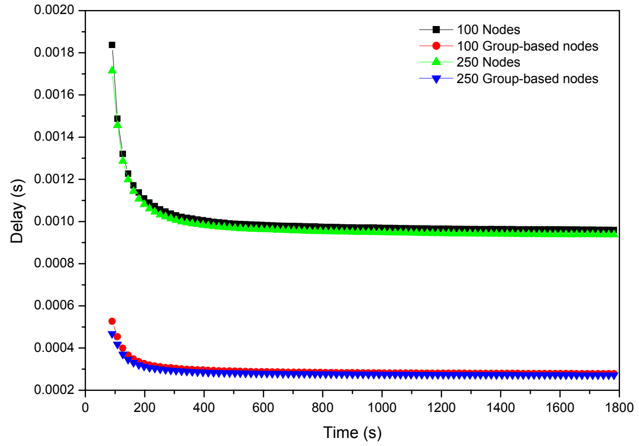


Fig. 16. Retardo medio en topologías móviles y con errores utilizando el protocolo AODV.

En la Fig. 17. tenemos el retardo medio a nivel MAC que sufren las topologías fijas. En el caso de 250 nodos obtenemos un retardo alrededor de los 0.00092 segundos que pasa a valer 0.00025 segundos en el caso de la arquitectura de 250 nodos basada en grupo, aportando una mejora del 73%. En el caso de 100 nodos la mejora que aporta la arquitectura basada en grupos es prácticamente cero, podemos ver las dos medidas se encuentran alrededor de los 0.00026 segundos.

En el caso que exista movilidad y fallos en los nodos (ver Fig. 18) estos datos ya cambian. Se puede observar el caso de las topologías de 100 nodos, tenemos un retardo medio igual a 0.000268 segundos una ver ha convergido la arquitectura clásica que disminuye a 0.000262 segundos en el caso del escenario basado en grupos. En el caso de 250 nodos la mejora no es tan alta, pasamos de 0.000262 segundos a 0.000260 segundos.

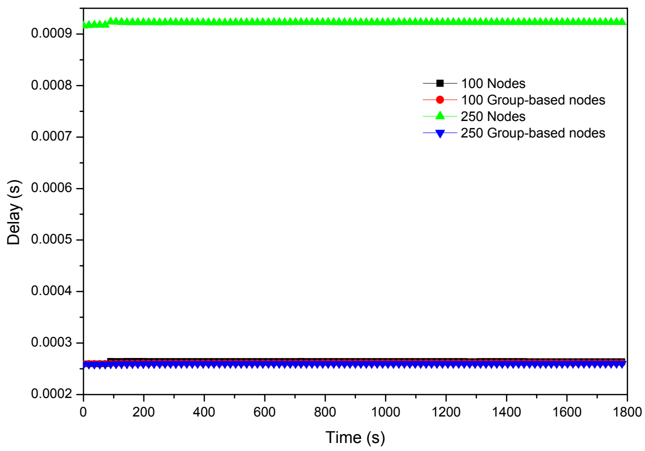


Fig. 17. Retardo medio en topologías fijas utilizando el protocolo OLSR.

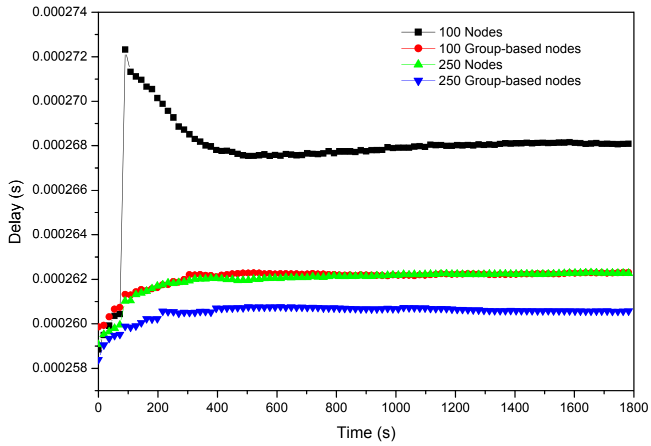


Fig. 18. Retardo medio en topologías móviles y con errores utilizando el protocolo OLSR.

IV.5. *Tráfico de datos MANET.*

Si observamos la Fig. 19. y Fig. 20. vemos que el tráfico MANET a nivel de aplicación que circula por la red también es menor cuando poseemos arquitecturas basadas en grupos. En ambas figuras vemos que conforme aumenta el número de nodos el tráfico disminuye, esto se debe a que la existencia de más nodos aporta que haya más nodos actuando como routers y por tanto tendremos una mayor probabilidad de éxito que el paquete llegue al destino.

En la Fig. 19. la topología de 100 nodos basada en grupos tiene una mejora de un 77% respecto a la topología de 100 nodos, en cambio en mejora disminuye conforme aumenta el número de nodos, en el caso de 250 nodos esta mejora es de un 60%.

Este comportamiento varía si la topología es móvil y con errores (ver Fig. 20.) en este caso la topología de 100 basada en grupos sigue teniendo una mejora respecto a la 100 nodos del 77%, en cambio esta mejora aumenta al 80% en el caso de la topología de 250 nodos basada en grupos.

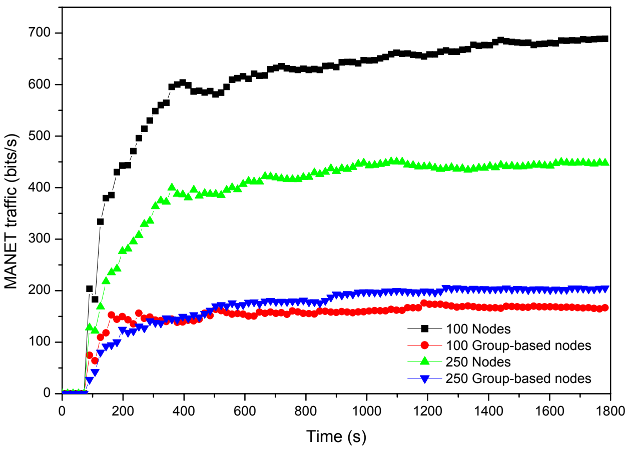


Fig. 19. Tráfico MANET en topologías fijas utilizando el protocolo DSR.

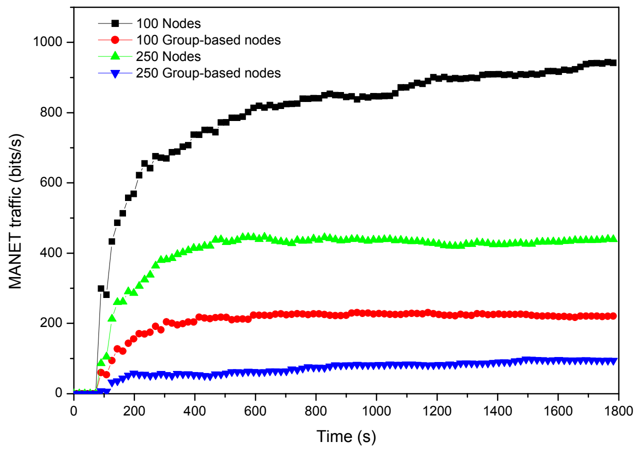


Fig. 20. Tráfico MANET en topologías móviles y con errores utilizando el protocolo DSR.

En la Fig. 21. y Fig. 22. está representada la evolución en media que sufre el tráfico MANET a nivel de aplicación para las diferentes topologías simuladas. El tráfico medio MANET en la topología de 100 nodos fijos tiene un valor medio de unos 600 bits/s, en cambio este tráfico disminuye en un 70% situándose alrededor de los 180 bits/s. Para la topología de 250 nodos ocurre el mismo fenómeno pero en este caso pasados de tener unos 480 Kbits/s a los 50 Kbits/s en el escenario basado en grupos, mejorando en un 90%. Con estos datos podemos ver el grado de escalabilidad que poseen las arquitecturas basadas en grupos (ver Fig. 21.).

En la Fig. 22. tenemos la misma simulación pero en este caso para topologías móviles y con errores. Aquí además de ver el grado de mejora, en este caso no tan relevante como en el anterior, lo que cabe destacar es la rapidez de convergencia que poseen las arquitecturas basadas en grupos. Observamos que en el caso de 100 nodos el tráfico no llega a converger hasta aproximadamente los 1400 segundos, en cambio para el mismo número de nodos basados en grupos está convergencia ya se da en el instante igual a 200 segundos. Lo mismo ocurre en la topología de 250 nodos, en este caso la arquitectura clásica converge a los 600 segundos mientras que la basada en grupos converge en torno a los 180 segundos.

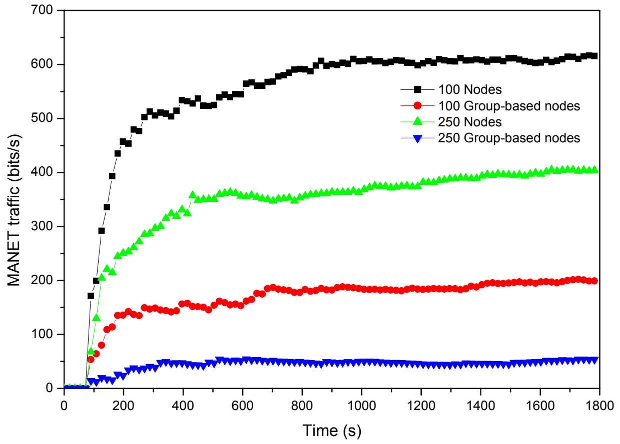


Fig. 21. Tráfico MANET en topologías fijas utilizando el protocolo AODV.

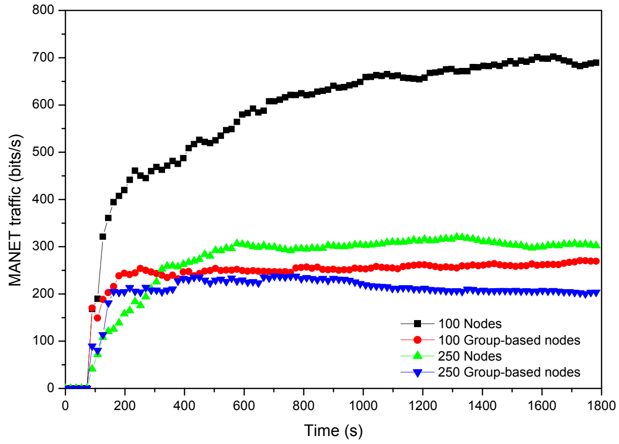


Fig. 22. Tráfico MANET en topologías móviles y con errores utilizando el protocolo AODV.

Si analizamos el tráfico MANET medio cuando utilizamos OLSR, vemos que en la Fig. 23. el tráfico medio en la topología de 100 nodos es igual a 700 bits/s, este tráfico disminuye hasta 180 bits/s cuando la topología esta basada en grupos, obteniendo un mejora del 75%. En el caso de 250 nodos está mejora disminuye a un 51%. En este caso tenemos un tráfico MANET alrededor de 450 bits/s cuando no hay grupos, que disminuye a 220 bits/s cuando la topología está basada en grupos.

Cuando tenemos escenarios móviles y con errores debemos de observar la Fig. 24. Al igual que ocurría con el protocolo AODV vemos que las topologías clásicas poseen un tiempo de convergencia mayor al de las arquitecturas basadas en grupo. En este caso la topología de 100 nodos nunca llega a converger dentro del periodo simulado. En el escenario de 250 nodos la red empieza a ser estable a partir del instante 1200 segundos. La mejora introducida en las topologías de 250 nodos es muy similar a la obtenida en los escenarios fijos (alrededor del 51%). El tráfico medio MANET que se obtiene en la topología de 100 nodos está en torno a los 900 bits/s. Para la topología basada en grupos este tráfico tiene un valor aproximado de 215 bits/s (76% de mejora).

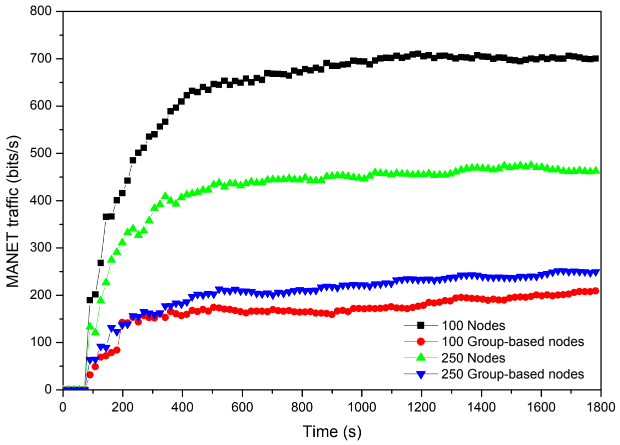


Fig. 23. Tráfico MANET en topologías fijas utilizando el protocolo OLSR.

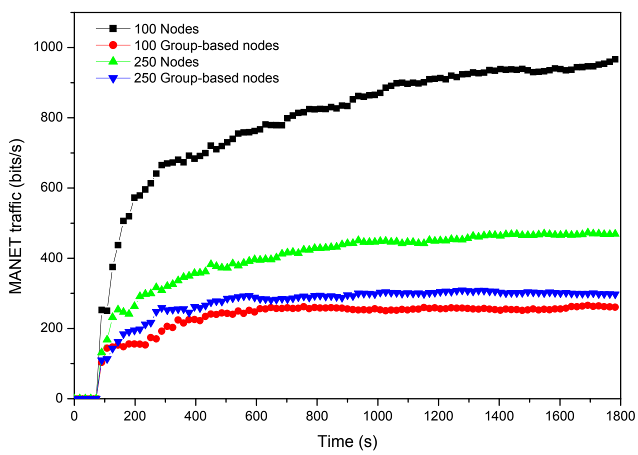


Fig. 24. Tráfico MANET en topologías móviles y con errores utilizando el protocolo OLSR.

IV.6. *Throughput medio.*

Si observamos el throughput medio que circula por la red cuando se está ejecutando el protocolo DSR (ver Fig. 25. y Fig. 26.) podemos ver que en las arquitecturas basadas en grupos es muy inferior al obtenido en una arquitectura normal. En la Fig. 25., para las arquitecturas de 100 nodos pasamos de tener un throughput de 225 Kbits/s a 100 Kbits/s en la arquitectura basada en grupos, obteniendo una mejora del 56%. En las topologías de 250 nodos tenemos unos 460 Kbits/s de throughput en la arquitectura clásica y 190 Kbits/s en la basada en grupos, lo que nos aporta una mejora del 58%. Además al comparar ambas figuras (Fig. 25. y Fig. 26.) obtenemos que el throughput medio en arquitecturas basadas en grupo poseen una variación muy pequeña respecto a si tenemos un escenario fijo o móvil, fenómeno que no ocurre en la otra arquitectura.

La mejora obtenida es de bastante importante, podemos ver en la Fig. 26. que en el punto 1200 segundos el throughput medio obtenido en la topología de 250 nodos basada en grupos es similar al throughput obtenido en la topología normal de 100 nodos. Aspecto que nos muestra una respuesta muy buena asociada a la escalabilidad de la red.

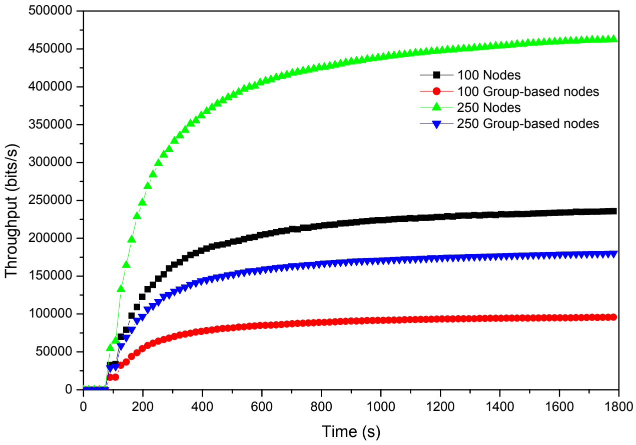


Fig. 25. Throughput medio en topologías fijas utilizando el protocolo DSR.

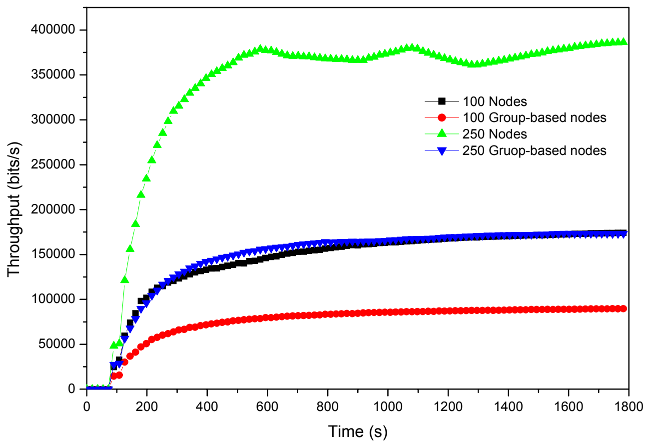


Fig. 26. Throughput medio en topologías móviles y con errores utilizando el protocolo DSR.

En la Fig. 27. tenemos el throughput medio consumido para las topologías fijas cuando se está ejecutando el protocolo AODV. Para el escenario de 100 nodos tenemos que el throughput medio simulado es igual a 200 Kbits/s en cambio cuando esa topología está basada en grupos tenemos unos 120 Kbits/s, obteniendo así una mejora del 40%. Para el caso de 250 nodos, tenemos un throughput medio de 425 Kbits/s en el escenario normal y 225 Kbits/s en el basado en grupos, aportando una mejora del 47%.

En la Fig. 28. tenemos la misma medida pero para topologías móviles y con errores. En este caso la mejora obtenida por la topología basada en grupos en el caso de 100 nodos disminuye y posee un valor igual a 37%. En cambio para las topologías de 250 nodos no varía.

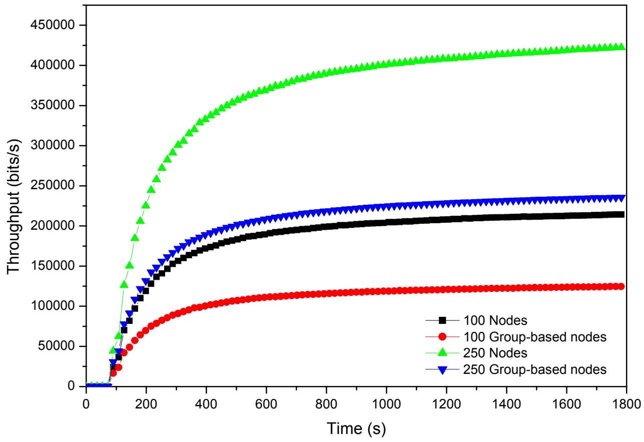


Fig. 27. Throughput medio en topologías fijas utilizando el protocolo AODV.

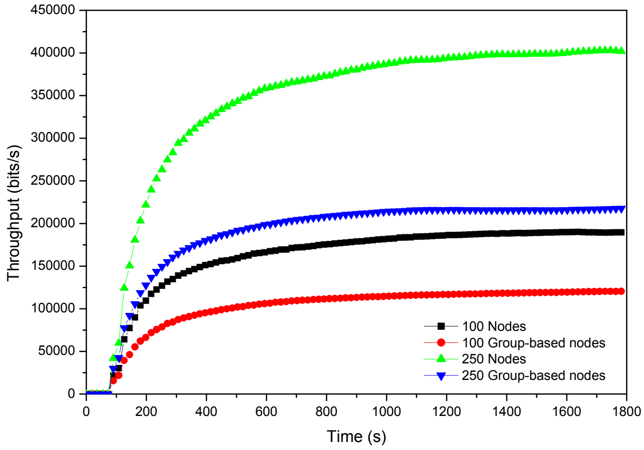


Fig. 28. Throughput medio en topologías móviles y con errores utilizando el protocolo el protocolo AODV.

El throughput medio consumido en las topologías fijas ejecutando el protocolo de encaminamiento OLSR se observa en la Fig. 29. Cuando tenemos un escenario con 250 nodos obtenemos un throughput igual a 550 Kbits/s que se ve disminuido a 250 Kbits/s en el caso de utilizar una arquitectura basada en grupos, mejorando la red en un 54%. Para la topología de 100 nodos clásica el throughput simulado es igual a 325 Kbits/s que pasa a ser de unos 125 Kbits/s en el caso del escenario basado en grupos (mejora del 61%).

Cuando las topologías son móviles y con errores (ver Fig. 30.) el throughput medio ya no es tan estable como en el caso anterior. Aunque como se puede observar las mejoras son muy similares. En el caso de 250 nodos tenemos una mejora del 52% y cuando tenemos 100 nodos esta mejora aumenta hasta el 60%.

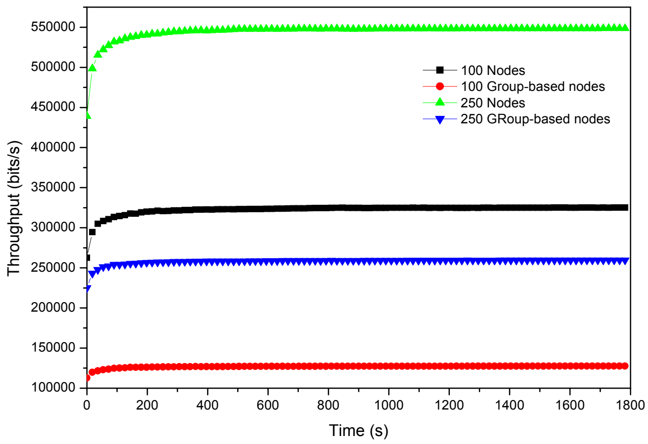


Fig. 29. Throughput medio en topologías fijas utilizando el protocolo OLSR.

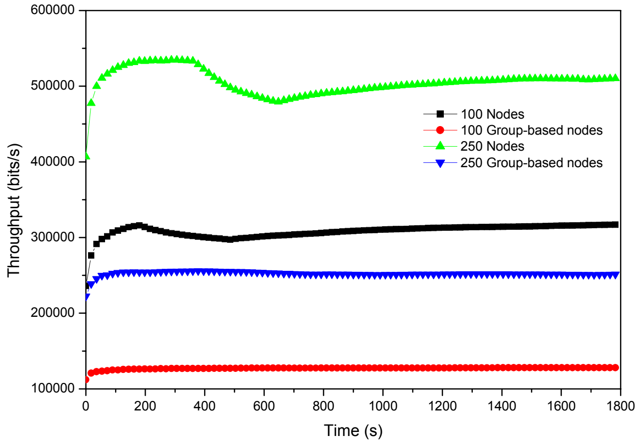


Fig. 30. Throughput medio en topologías móviles y con errores utilizando el protocolo el protocolo OLSR.

IV.5. *Comparativa de los sistemas basados en grupos.*

En este punto vamos a comparar los diferentes protocolos simulados (DSR, AODV y OLSR) para ver cual de ellos posee mejores características en topologías basadas en grupos. Para ello se presentan las siguientes simulaciones donde se observa el comportamiento de cada protocolo.

El retardo medio a nivel MAC en topologías fijas basadas en grupos se observa en la Fig. 31. El retardo medio para todos los protocolos en topologías de 100 y 250 nodos, siempre es inferior a 0.001 segundos (cuando la red se estabiliza), una característica importante que demuestra el buen comportamiento de la agrupación de nodos. Si nos fijamos en la gráfica obtenido podemos ver que de los tres protocolos el que peor comportamiento aporta es el DSR con 100 nodos, en cambio el mejor es el OLSR con 250 nodos. El protocolo OLSR tiene aproximadamente el mismo retardo (0.001 segundos) para los escenarios de 100 y 250 nodos y además es muy estable.

En la Fig. 32. se observa la misma medida pero en este caso para topologías móviles y con errores. Todos los protocolos poseen un retardo inferior a 0.001 segundos una vez la red converge, igual que ocurría en las topologías fijas. El protocolo AODV es el de peor comportamiento. El protocolo OLSR sigue siendo el más estable.

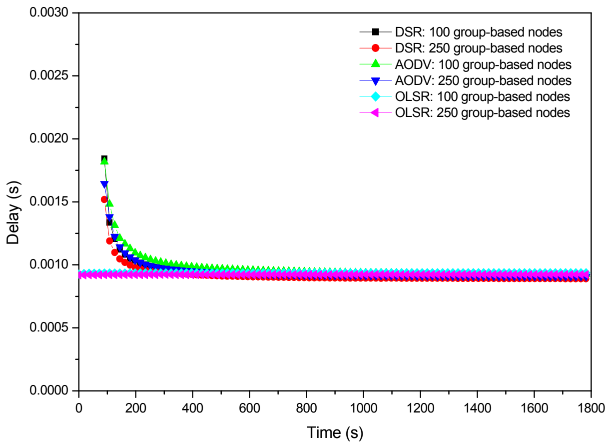


Fig. 31. Comparativa del retardo medio en topologías fijas.

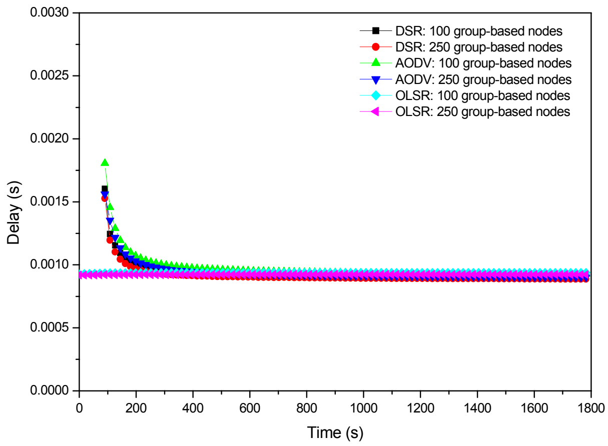


Fig. 32. Comparativa del retardo medio en topologías móviles y con errores.

En la Fig. 33. se muestra el retardo medio a nivel que aplicación que poseen los diferentes protocolos simulados basados en grupos para las topologías fijas de 100 y 250 nodos. El protocolo más inestable y él que mayor retardo introduce es el protocolo AODV en el escenario de 100 nodos. Tiene picos que llegan a los 0.45 segundos y se estabiliza en torno a los 1700 segundos, con un valor aproximado de 0.15 segundos. El resto de protocolos poseen un retardo muy bajo. Como hemos podido observar en las simulaciones anteriores el protocolo más estable es el OLSR.

En la siguiente simulación (ver Fig. 34.) tenemos la misma medida anterior pero en este caso para topologías móviles. El protocolo DSR es el que peor respuesta aporta antes de la convergencia, teniendo un retardo de pico de 1.2 segundos, una vez se estabiliza la red el AODV es que mayor retardo a nivel de aplicación introduce (intervalo de 0.1 a 0.15 segundos). OLSR es el que menor retardo aporta, teniendo un valor muy cercano a cero. Además de estás características, tenemos que observar el comportamiento que poseen estos protocolos ante la movilidad y los fallos, el AODV se trata del protocolo menos estable, es cambio el OLSR aporta mayor estabilidad.

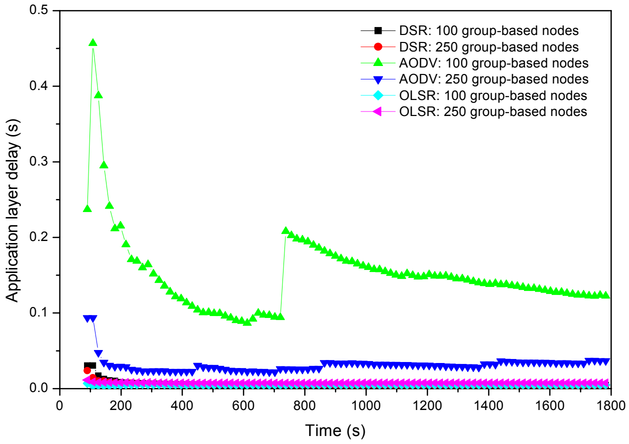


Fig. 33. Comparativa del retardo medio a nivel de aplicación en topologías fijas.

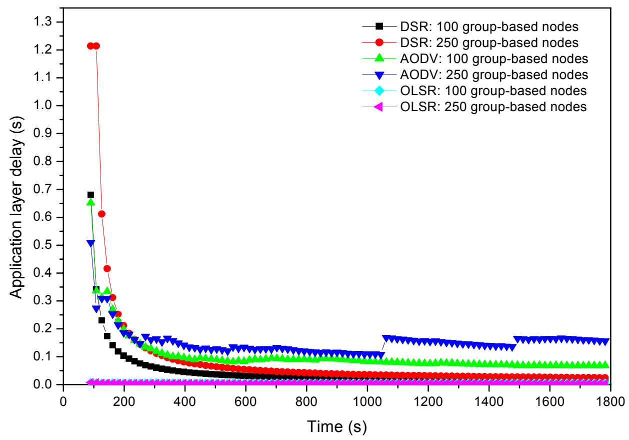


Fig. 34. Comparativa del retardo medio a nivel de aplicación en topologías móviles y con errores.

Ahora vamos a comentar el comportamiento que poseen los protocolos de enrutamiento basados en grupos en las topologías fijas (ver Fig. 35). Podemos observar que el protocolo que introduce mayor tráfico de encaminamiento es el AODV, posee un tráfico alrededor de los 120 Kbit/s en la topología de 250 nodos y 56 Kbits/s en la de 100 nodos. El protocolo DSR posee similar comportamiento pero aporta un poco menos de tráfico de encaminamiento. El OLSR es el protocolo que mejor comportamiento posee a nivel de red, se trata del protocolo más estable y además es él que menor tráfico inyecta, tenemos unos 64 Kbits/s en el caso de 250 nodos y unos 28 Kbits/s en el escenario de 100 nodos.

Si analizamos la topología móvil y con errores (Fig. 36) observamos que el comportamiento es muy similar al presentado en la Fig. 35. En este caso disminuye ligeramente el tráfico de encaminamiento en todos los protocolos. El AODV sigue siendo el de peor comportamiento porque introduce mayor cantidad de tráfico a nivel de red. OLSR sigue siendo el protocolo más estable y de menor carga frente a la movilidad y los fallos de la red. En la Fig. 36 cabe destacar que el protocolo DSR en arquitecturas basadas en grupo es el menos estable frente a las características de movilidad y errores que poseen las topologías simuladas.

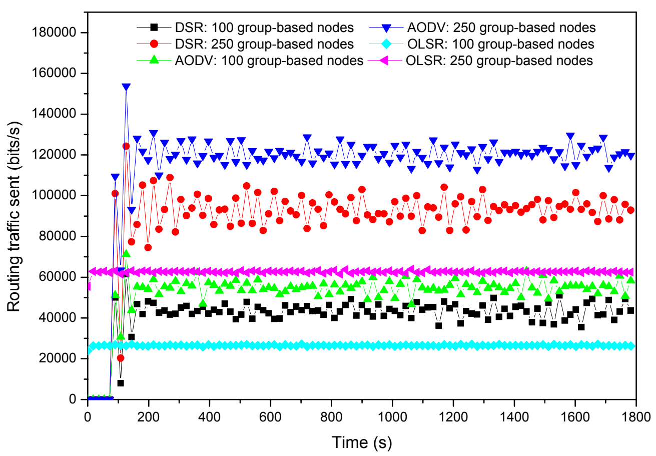


Fig. 35. Comparativa del tráfico de encaminamiento enviado en topologías fijas.

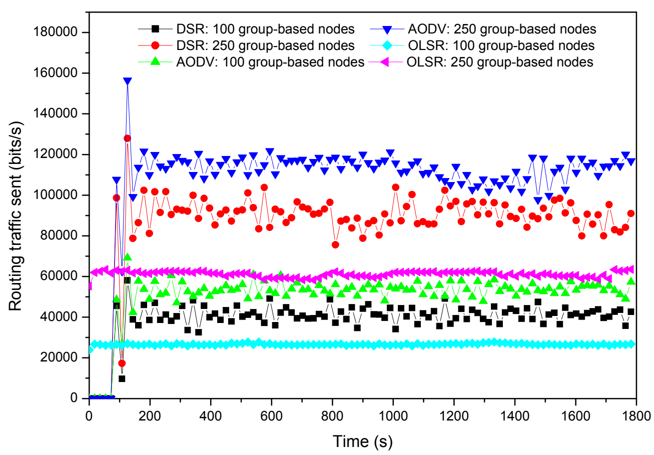


Fig. 36. Comparativa del tráfico de encaminamiento enviado en topologías móviles y con errores.

Antes de analizar el comportamiento que poseen los diferentes protocolos respecto al tráfico de encaminamiento recibido, a primera vista (Fig. 37 y Fig. 38) y comparándolo con el tráfico de encaminamiento enviado (Fig. 35 y 36) vemos que tienen un patrón muy similar y lo único que varía es la cantidad de tráfico. Esto se debe a que un nodo posee diferentes vecinos y por tanto recibe información de encaminamiento de varias fuentes en cambio el sólo envía de una fuente, él mismo.

Para comparar el tráfico de encaminamiento recibido debemos observar las Fig. 37 y 38. Podemos ver que en las topologías fijas (ver Fig. 37) el protocolo AODV introduce mayor tráfico de encaminamiento (alrededor de los 250 Kbit/s en la topología de 250 nodos y 135 Kbits/s en la de 100 nodos. El protocolo DSR aporta unos 200-190 Kbits/s en el escenario de 250 nodos y unos 100 Kbits/s en el escenario de 100 nodos. En cambio, el OLSR es el protocolo más estable y además él que menor tráfico de nivel de red aporta, tenemos unos 145 Kbits/s en el caso de 250 nodos y unos 70 Kbits/s en el escenario de 100 nodos.

Si analizamos las topologías móviles y con errores (Fig. 38). En este caso disminuye ligeramente el tráfico de encaminamiento debido principalmente por las características de movilidad y fallos. El AODV es él de peor comportamiento. OLSR sigue siendo el protocolo más estable y él de menor carga. El DSR en arquitecturas basadas en grupo es el menos estable.

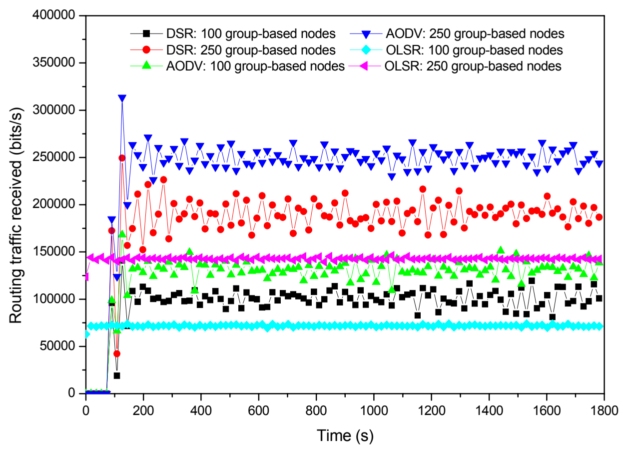


Fig. 37. Comparativa del tráfico medio recibido en topologías fijas.

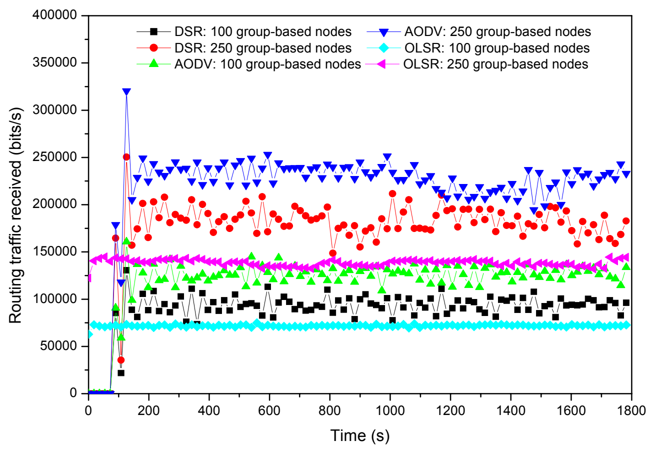


Fig. 38. Comparativa del tráfico medio recibido en topologías móviles y con errores.

Si analizamos el throughput medio consumido en las topologías fijas (Fig. 39.) obtenemos que el protocolo que menor throughput introduce a la red es el protocolo DSR (90 Kbits/s con 100 nodos y 170 Kbits/s con el escenario de 250 nodos). El protocolo que aporta un throughput medio más estable es el OLSR. Se puede observar que una vez converge el protocolo AODV en la topología de 100 nodos el throughput medio es igual al aportado por el protocolo OLSR, por tanto podemos indicar que los protocolos AODV y OLSR introducen el mismo throughput medio una vez la red es estable, pero el OLSR posee un tiempo de convergencia inferior.

Con topologías con movilidad y posibles errores (ver Fig. 40.) los resultados son muy similares a los anteriores, ligeramente se obtiene un throughput inferior. El protocolo que menor throughput introduce es el DSR. El protocolo AODV introduce un poco menos de carga mientras se estabiliza la red, pero esta carga tiende a ser muy similar a la introducida por el protocolo OLSR. Este último protocolo sigue siendo él más estable.

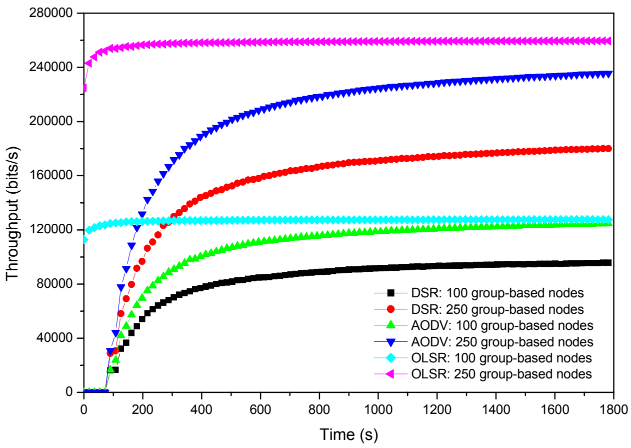


Fig. 39.Comparativa del throughput medio consumido en topologías fijas.

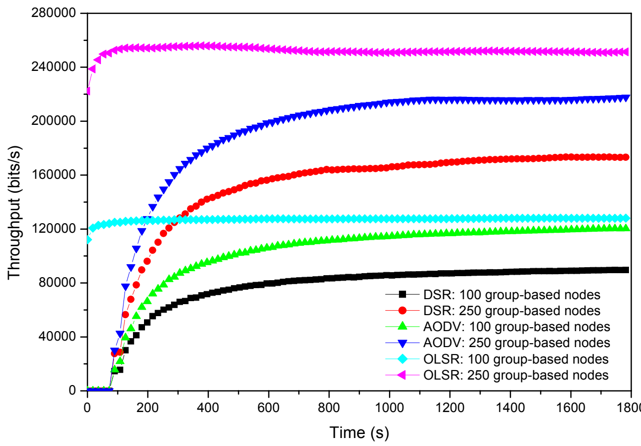


Fig. 40. Comparativa del throughput medio consumido en topologías móviles y con errores.

Vamos a ver el comportamiento de los protocolos ante el tráfico MANET. Cuando tenemos topologías fijas (ver Fig. 41) el protocolo que aporta menor carga el AODV en el caso de 250 nodos (aproximadamente 50 bits/s). Cuando simulamos los escenarios de 100 nodos podemos ver que aproximadamente todos los protocolos poseen el mismo comportamiento, están dentro del intervalo de 150 a 180 bits/s, siendo el DSR el mejor en este tipo de escenario.

Cuando existe movilidad en la red debemos de observar la Fig. 42 en este gráfico podemos ver que el protocolo que menor carga aporta es el DSR con unos 90 bits/s aproximadamente (topologías de 250 nodos). Para el caso de 100 nodos todos los protocolos poseen un comportamiento similar, todos se sitúan entre los 225 y 275 bits/s, siendo el DSR el que menor tráfico MANET introduce. Cuando tenemos movilidad y fallos en la red la cantidad de tráfico MANET aumenta aproximadamente un 20% en el mejor de los casos.

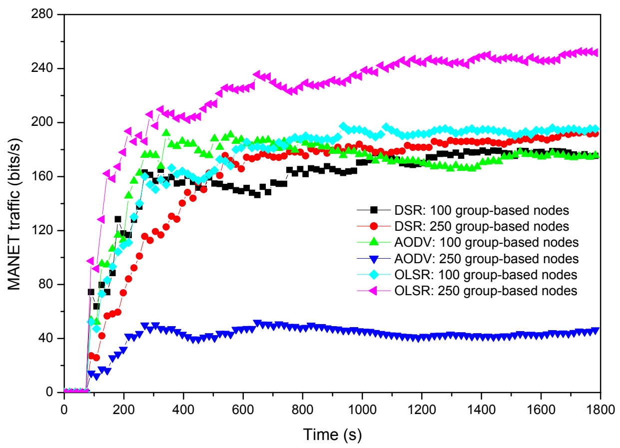


Fig. 41. Comparativa del tráfico MANET medio en topologías fijas.

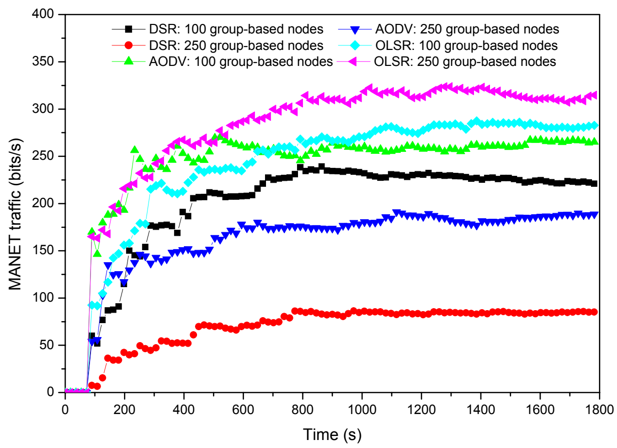


Fig. 42. Comparativa del tráfico MANET medio en topologías móviles y con errores.

En las siguientes comparativas (Fig. 43, 44, 45 y 46) sólo se analizan los protocolos DSR y AODV, debido a que estas características son propias de los protocolos reactivos.

En la Fig. 43 se observa el número medio de saltos por ruta para los diferentes protocolos en topologías físicas. El protocolo DSR posee un número medio de saltos cercano a 5 en el escenario de 250 nodos según va convergiendo la red. El número de saltos medio en el caso de 100 nodos es ligeramente inferior. En el caso del protocolo AODV el número medio de saltos disminuye considerablemente, en el caso de 250 nodos tenemos un número medio de saltos 3.25 y en el caso de 100 nodos 2.75. Además se puede ver que el tiempo de convergencia en el caso del protocolo AODV es muy inferior al DSR, en el instante 300 segundos el AODV ya es estable, en cambio el DSR termina la simulación y aún no es estable.

Cuando analizamos la misma medida en el caso de topologías móviles y con errores (ver Fig. 44) vemos que el comportamiento es muy similar al que se obtiene en el caso de topologías fijas. Con ello la conclusión que extraemos es que el número medio de saltos por ruta en topologías con un elevado número de nodos posee una dependencia más fuerte del tipo de protocolo a nivel de red que del número de nodos.

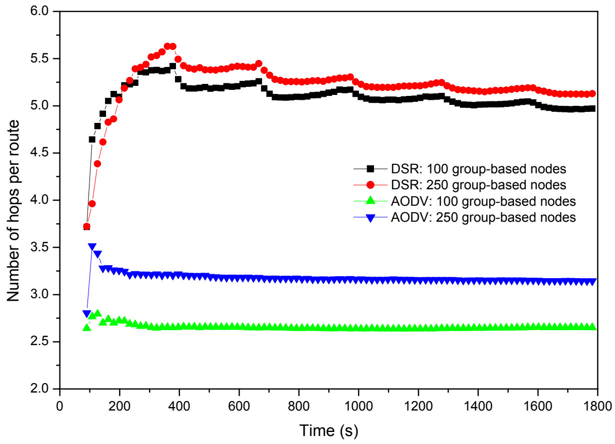


Fig. 43. Comparativa del número medio de saltos por ruta en topologías fijas.

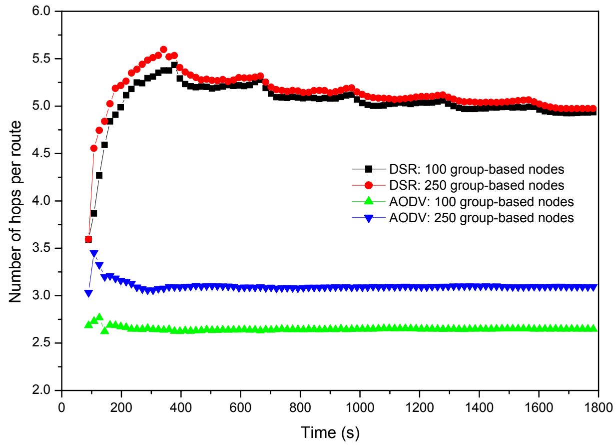


Fig. 44. Comparativa del número medio de saltos por ruta en topologías móviles.

Por último vamos a analizar el único parámetro simulado donde existe un peor comportamiento en arquitecturas basadas en grupo que en arquitecturas clásicas.

El parámetro route requests sent simulado en las Fig. 45 y Fig. 46 para topologías fijas y móviles con errores, respectivamente, vemos que posee un comportamiento muy similar en ambos escenarios. El protocolo AODV introduce un mayor número de route requests sent (aproximadamente 860) en topologías de 250 nodos, en cambio este parámetro disminuye hasta los 330 en topologías de 100 nodos, obteniendo una relación aproximada donde el número de route requests sent en el protocolo AODV es igual al número de nodos multiplicado por 3.3.

En el caso del protocolo DSR este número de route requests sent es igual a 730 en el escenario de 250 nodos y 190 en el caso de tener 100 nodos. Como se puede observar el protocolo DSR introduce menos route requests sent que el AODV. En este caso solo existe una relación lineal entre el número de nodos y la cantidad de route requests sent.

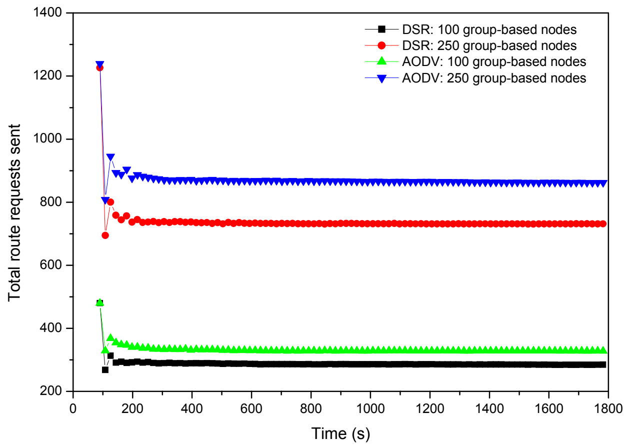


Fig. 45. Comparativa del número medio de route requests sent en topologías fijas.

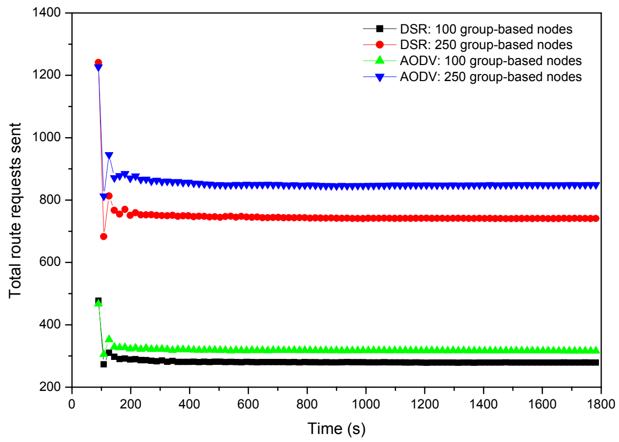


Fig. 46. Comparativa del número medio de route requests sent en topologías móviles.

IV.6. *Resumen de los protocolos analizados.*

En este punto vamos a ver de manera compacta los beneficios de usar sistemas basados en grupo en redes ad hoc. Hemos simulados los protocolos DSR, AODV y OLSR, con y sin grupos, y los resultados que hemos extraído es que los sistemas basados en grupos aportan un mejor rendimiento global a la red.

En la tabla 1 podemos observar un resumen donde se indica el porcentaje de mejora que aporta a la red cuando se están utilizando arquitecturas basadas en grupo.

	Topología fija (100 nodos)	Topología fija (250 nodos)	Topología móvil y error (100 nodos)	Topología móvil y error (250 nodos)
Retardo medio a nivel aplicación DSR	75%	83%	5%	5%
Tráfico de encaminamiento recibido DSR	60%	60%	46%	55%
Throughput medio DSR	56%	59%	48%	55%
Retardo medio a nivel aplicación AODV	70%	70%	67%	75%
Tráfico de encaminamiento recibido AODV	39%	43%	32%	43%
Throughput medio AODV	40%	47%	37%	47%
Retardo medio a nivel aplicación OLSR	60%	76%	64%	60%
Tráfico de encaminamiento recibido OLSR	61%	50%	61%	50%
Throughput medio OLSR	54%	61%	52%	60%

Tabla 1. Resumen de los porcentajes de mejora que aportan las topologías basadas en grupos.

En este estudio sobre arquitecturas basadas en grupo, hemos obtenido más datos [40] [41], que no han sido incluidos en esta memoria porque creemos que los mostrados son los más importantes. En la tabla 2 se muestra los mejores y peores protocolos para cada uno de los parámetros analizados.

	Mejor en topología fija	Mejor en topología móvil	Peor en topología fija	Peor en topología móvil
Retardo a nivel MAC	OLSR	OLSR	DSR	AODV
Throughput medio	DSR	DSR	AODV y OLSR	AODV y OLSR
Tráfico MANET	AODV	DSR	OLSR	OLSR
Tráfico de encaminamiento enviado	OLSR	OLSR	AODV	AODV
Tráfico de encaminamiento recibido	OLSR	OLSR	AODV	AODV
Retardo a nivel de aplicación	DSR y OLSR	OLSR	AODV	AODV
Número medio de saltos por ruta	AODV	DSR	AODV	DSR
Envío de mensajes RRQ	DSR	AODV	DSR	AODV

Tabla 2. Relación de protocolos según varios parámetros.

El porcentaje de mejora más alto, cuando usamos topologías basadas en grupo con nodos fijos, fue para el protocolo DSR en la simulación donde se está recogiendo información sobre el retardo medio a nivel de aplicación. Por otro lado, en ese mismo parámetro con topologías móviles y con errores, el protocolo DSR es el que peor comportamiento aporta y por tanto el menor porcentaje de mejora.

Podemos observar que existe un mayor porcentaje de mejora en topologías fijas cuando hay más nodos en la arquitectura, pero cuando tenemos una topología móvil, la mejora es mayor si existe un menor número de nodos. También hemos visto que cuando un protocolo de enrutamiento posee mejores características en la topología fija basada en grupos, sigue siendo el mejor en la topología móvil basada en grupos.

Por otra parte, se observa que un protocolo de encaminamiento, que es el mejor (o peor) en una topología fija basada en grupos, no puede ser el mejor (o peor) en la topología móvil basada en grupos. El protocolo de encaminamiento que aporta mejores características es el OLSR, en cambio el protocolo AODV es el de peor comportamiento.

V. ARQUITECTURA PROPUESTA.

V.1. *Descripción de la arquitectura propuesta.*

La propuesta se basa en la creación de una arquitectura basada en grupos de nodos que poseen la misma funcionalidad dentro de la red. Cada grupo tiene un nodo central que delimita la zona en la que estarán los nodos que se encuentren en su grupo, pero su funcionalidad es la misma que la del resto de los nodos. Cada nodo tiene un identificador de nodo (nodeID) que es único en su grupo. El primer nodo de la red adquiere un identificador de grupo (groupID) bien de manera manual, usando un GPS (Global Positioning System), utilizando un sistema de localización inalámbrica o por otros medios [42].

Los nuevos nodos que se unan sabrán el identificador de grupo de sus nuevos vecinos. Los nodos frontera son, físicamente, los nodos que delimitan el grupo. Cuando hay un evento en un nodo, este evento se envía a todos los nodos de su grupo con el fin de tomar las medidas adecuadas. Los nodos frontera tienen conexiones con otros nodos frontera de los grupos vecinos y se utilizan para enviar información a otros grupos o para recibir información de otros grupos y distribuirla en el interior de su grupo.

Como es necesario un protocolo de encaminamiento rápido hemos elegido SPF (Shortest Path First) como algoritmo enrutamiento [43] para enviar información de ruta, pero éste se puede cambiar por otro protocolo de encaminamiento dependiendo de las características de la red. Cuando la información va dirigida hacia un nodo del mismo grupo se encamina usando el nodeID. Cada nodo ejecuta el algoritmo SPF localmente y selecciona el mejor camino hacia ese destino basándose en una métrica. Pero cuando la información tiene que ser enviada a otros grupos, se realiza directamente a través del nodo frontera más cercano al grupo destino a través del groupID. Cuando un nodo del grupo destino recibe información, está se encamina a todos los nodos en su grupo usando Reverse Path Forwarding Algorithm [44].

Los enlaces entre nodos frontera de diferentes grupos se establecen principalmente en función de su posición, pero, en el caso de que existan múltiples posibilidades, los vecinos se seleccionan en función de su capacidad λ que será explicada en el punto siguiente. Para establecer los límites del grupo, podemos considerar dos opciones: i) limitar el diámetro del grupo de un número máximo de saltos (por ejemplo, 30 saltos, como el número máximo de saltos que dispone el trace route), y ii) establecer los límites de la zona que va a ser cubierta. La Fig. 47 muestra la arquitectura propuesta de la topología.

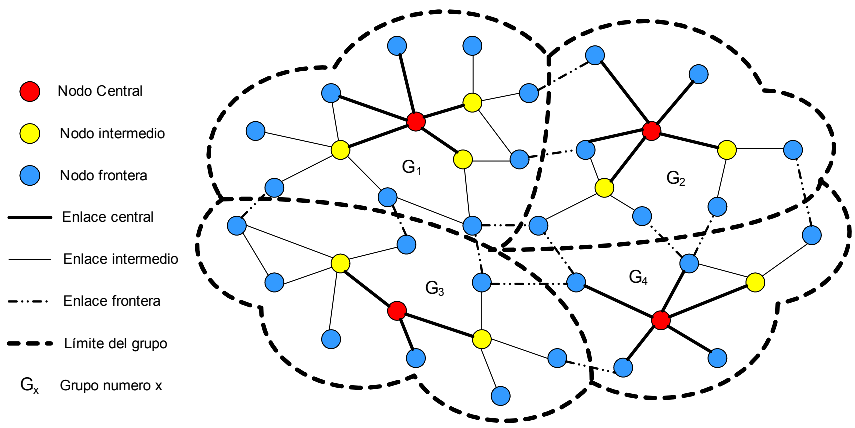


Fig. 47. Topología de la arquitectura propuesta.

V.2. *Modelo analítico y selección de vecinos.*

Cada nodo dispone de tres parámetros que lo caracterizan, que son el nodeID, el groupID y λ. El parámetro λ determina la capacidad del nodo, que depende del ancho de banda de súbida y de bajada del nodo (en Kbps), del número de conexiones disponibles (Available_Con), del número máximo de enlaces (Max_Con), del porcentaje de carga disponible y del consumo de energía. Todos estos parámetros son tenidos en cuenta para determinar el mejor nodo al cual deben conectarse. El nodo que disponga de mayor λ será el mejor candidato para conectarse. En la ecuación (1) podemos ver la expresión que determina λ.

$$\lambda = \frac{\left(BW_{up} + BW_{down}\right) \cdot Available\_Con \cdot L + K_2}{Max\_Con} \cdot \sqrt{1 - \frac{E^2}{K_1}} \qquad (1)$$

Donde L determina la carga disponible y E es la energía consumida. Estos valores varían entre 0 y 100. E = 0 indica que el nodo dispone de toda su energía, por tanto el parámetro λ es igual a 0 si E = 100, que indica que el nodo no dispone de energía.

K_1 define el valor mínimo de energía restante en un nodo, apropiada para ser seleccionado como vecino. K_2 da diferentes valores de λ, que van desde 0 en el caso de L=0 ó Available_Con=0. Hemos considerado K_2=100 para conseguir λ en un rango de valores deseados. En la Fig. 48 se muestran los valores del parámetro λ cuando el número máximo de enlaces para un nodo son 16 y un valor de ancho de banda máximo es de 2 Mbps, en función del número de enlaces disponibles para diferentes valores de energía disponible en un nodo. Hemos fijado la carga del nodo en un 50%. La Fig. 49 muestra el valor del parámetro λ cuando el número de máximo de enlaces para un nodo es 16 y todos disponen del mismo numero de enlaces disponibles (Available_con=6) en función de la energía disponible en el nodo para diferentes valores de ancho de banda. En este caso los nodos disponen de una carga del 80%. Se puede observar que conforme se va consumiendo energía en el nodo el valor de λ disminuye lentamente, pero entorno al 80% existe una caída muy abrupta, a raíz de esto vemos que un nodo tendrá mayor probabilidad de ser elegido como vecino si dispone de mayor energía. También se observa (ver Fig. 49) que es preferible un nodo que disponga de enlaces de mayor velocidad.

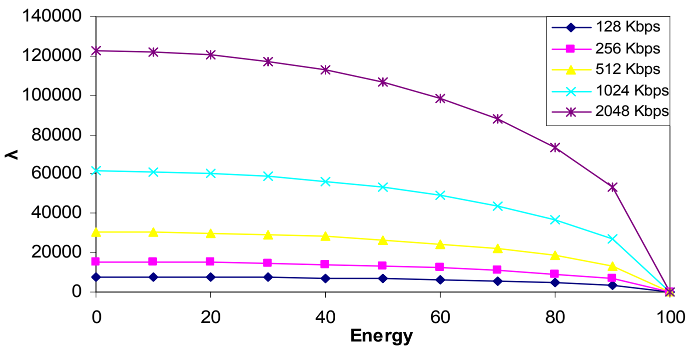


Fig. 48. Evolución del parámetro λ en función de la energía consumida del nodo.

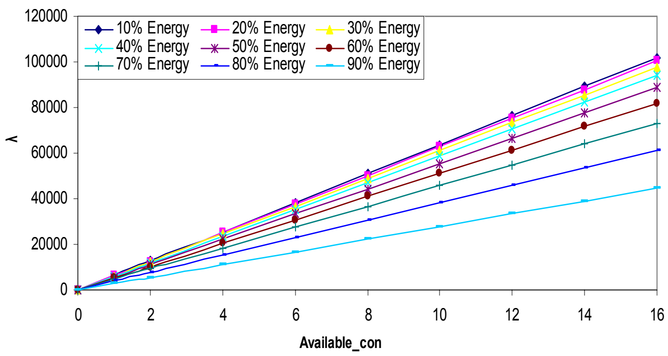


Fig. 49. Evolución del parámetro λ en función de las conexiones disponibles.

Nosotros definimos el coste de un nodo i-esimo directamente proporcional a T (el retardo de respuesta en milisegundos) e inversamente proporcional al parámetro λ de dicho nodo i-esimo. El coste se puede ver en la expresión (2).

$$C = \frac{T \cdot K_3}{\lambda} \qquad (2)$$

$K_3=10^3$ para obtener $C \geq 1$. La métrica para cada ruta esta basada en el número de saltos hasta el destino (r) y en el coste de los nodos (C_i) que atraviesa la información. En la ecuación (3) vemos como obtener la métrica.

$$metrica = \sum_{i=1}^{r} C_i \tag{3}$$

La métrica determina el mejor camino para encontrar un nodo destino.

Dada una red de nodos G = (V, λ, E), donde V es el conjunto de nodos, λ el conjunto de capacidades (λ(i) es la capacidad del nodo i y $\lambda(i) \neq 0 \ \forall$ nodo i) y E el conjunto de conexiones entre ellos. Sea k, un número finito de subconjuntos disjuntos de V, donde $V=\cup V_k$ y no existe ningún nodo común entre ellos ($\cap V_k=0$). Dado un nodo v_{ki} (sensor i del subconjunto k), no tendrá conexiones con sensores de su mismo subconjunto ($e_{ki-kj}=0 \ \forall$ Vk). Supongamos que n=|V| (número total de nodos de V) y k el número de subconjuntos de V, obtenemos la ecuación (4).

$$n = \sum_{i=1}^{k} | V_k | \tag{4}$$

Cada V_k tiene un nodo central, varios nodos intermedios y varios nodos frontera. Por tanto, para un solo grupo tenemos la expresión (5).

$$n = 1 + n_{\text{int}ermedio} + n_{frontera} \tag{5}$$

Los sensores de cada grupo son la suma de todos ellos. Por tanto, podemos describir la red completa como la suma de todos los sensores centrales más los intermedios más los sensores frontera tal como se indica en la expresión (6).

$$n = \sum_{i=1}^{k} | (n_{central} + n_{\text{int}ermedio} + n_{frontera})_k | = k + \sum_{i=1}^{k} (|n_{\text{int}ermedio}|)_k + \sum_{i=1}^{k} (|n_{frontera}|)_k \tag{6}$$

Por otro lado, el número de enlaces en toda la red m=|E| depende del número de grupos (k), del número de enlaces en cada grupo (k_m) y del número de enlaces entre nodos frontera. La ecuación (7) muestra el valor de m para una topología física.

$$m = \sum_{i=1}^{k} \left(k_l + \frac{1}{2} k_b \right) \tag{7}$$

Donde k_l es el número de enlaces dentro del grupo k y k_b es el número de enlaces entre grupos del grupo k.

VI. CONCLUSIONES.

VI.1. *Cumplimiento de los objetivos.*

El objetivo principal de esta tesina era demostrar que las redes basadas en grupos aportan un mejor rendimiento en la redes WAHSN convencionales. Como hemos podido observar mediante las simulaciones este objetivo lo hemos cumplido, ya que hemos visto que cuando poseemos diversos grupos en nuestra red tanto el tráfico de encaminamiento, el retardo, el throughput, etc. disminuye.

Otros de los objetivos que nos planteamos al realizar esta tesina fue conocer los diferentes tipos de redes y estudiar en profundidad aquellas que estén basadas en grupos. Este pequeño estudio se puede observar en el segundo punto, donde se habla de las arquitecturas basadas en grupos.

En el capitulo tres vimos uno de los objetivos que nos propusimos que era analizar cuales pueden ser las posibles aplicaciones reales donde se podrían aplicar arquitecturas basadas en grupos. En ese punto se pueden observar diferentes situaciones donde se podrían aplicar las topologías basadas en grupos.

Otro objetivo que hemos cumplido fue el de manejar el simulador de redes con él cual se han obtenido los resultados de esta tesina. Si no hubiéramos sido capaces de manejar este simulador esta tesina no tendría sentido.

Otro aspecto que hemos abordado es él de analizar qué protocolos poseen mejor respuesta según el parámetro que se estén analizando, una vez hemos visto que los protocolos poseen mejor comportamiento cuando funcionan sobre topologías basadas en grupos, en el punto cuatro hemos visto que porcentaje de mejoran aporta cada protocolo y cuales de ellos son mejores respecto a un parámetro.

Por último otro objetivo que planteamos al principio y que hemos cumplido era el de introducir una nueva arquitectura basada en grupos, que se ajuste de la mejor manera posible a las características propias de las redes WAHSN. Pensamos que la arquitectura que hemos presentado puede funcionar de una manera óptima en este tipo de redes, pero para poder afirmar con rotundidad está característica se necesita desarrollar con mayor profundidad dicha arquitectura, aspecto que se realizará en futuros trabajos.

VI.2. *Conclusiones sobre la tesina.*

A raíz de todas las simulaciones presentadas en el quinto punto se pueden extraer diferentes conclusiones. En primer lugar indicar que los protocolos estándar MANET en un principio no fueron diseñados para funcionar sobre redes de sensores inalámbricas propiamente dichas, pero el hecho de introducir tan poco tráfico de enrutamiento hace que sea posible utilizar estos protocolos sobre redes WAHSN.

Otro aspecto muy importante que no debemos olvidar, es que la utilización de este tipo de protocolos hace posible el manejo del protocolo IP. Una característica muy interesante, dando que actualmente se pretende que todo tenga conectividad utilizando el protocolo IP.

Además hemos podido observar que no todos los protocolos poseen el mismo comportamiento, característica que era de esperar, por ello hemos realizado esa comparativa donde se puede observar que protocolo posee un mejor comportamiento en cada dato simulado.

El porcentaje de mejora más alto, cuando usamos arquitecturas basadas en grupo con nodos fijos, fue para el protocolo DSR en la simulación donde se está recogiendo información sobre el retardo medio a nivel de aplicación. Por otro lado, en ese mismo parámetro con arquitecturas móviles y con errores, el protocolo DSR es el que peor comportamiento aporta y por tanto el menor porcentaje de mejora.

Podemos observar que existe un mayor porcentaje de mejora cuando hay más nodos en la arquitectura con nodos fijos, pero cuando tenemos nodos móviles, la mejora es mayor si existe un menor número de nodos. También hemos visto que cuando un protocolo de enrutamiento posee mejores características en el escenario fijo basado en grupos, sigue siendo el mejor en el escenario móvil basada en grupos.

Por otra parte, se observa que un protocolo de encaminamiento, que es el mejor (o peor) en una topología fija basada en grupos, no puede ser el mejor (o peor) en la topología móvil basada en grupos. El protocolo de encaminamiento OLSR es él que aporta mejores características, en cambio el protocolo AODV es él que peor comportamiento posee.

Como conclusión final debemos de quedarnos con, tal y cómo hemos podido apreciar a raíz de todas las simulaciones presentadas, vemos que los sistemas basados en grupos aportan un rendimiento mucho mayor que los sistemas convencionales, ya sea porque introducen menor tráfico de encaminamiento en la red, porque el retardo es mucho menor, etc.

VI.3. *Problemas encontrados y cómo se han solucionado.*

El primer problema que nos encontramos en esta tesina, fue el hecho de seleccionar el simulador. En un principio no sabíamos que simulador elegir, ya que ni NS2 ni OPNET nos daban la opción de crear grupos de una forma clara. Apostamos por OPNET porque disponíamos de licencias gracias al programa universitario que posee este mismo software y ya conocíamos su funcionamiento.

Una vez seleccionado el simulador de redes, nos pusimos manos a la obra para intentar crear grupos dentro de nuestra topología convencional. Este punto nos costo bastante tiempo debido a que el hecho de crear grupo dentro de una topología tal y como nosotros pensábamos no era trivial. Al final lo pudimos realizar gracias a un módulo que existía en las librerías del OPNET Modeler.

Otro pequeño problema que nos surgió fue cómo presentar los datos obtenidos para que aportaran la mayor cantidad de información al lector. Optamos por según el dato que quisiéramos presentar, primero mostrar el resultado comparando la topología convencional con la topología basada en grupos y después mostrar sólo la topología basada en grupos para ver que protocolo tenía un mejor comportamiento.

Por último, otro de los problemas que aparecieron en esta tesina fue el hecho de representar mediante un modelo matemático la arquitectura que se presenta en el quinto punto. Hasta el momento esta forma de describir arquitecturas o topologías de red no la había utilizado nunca. Gracias a la ayuda prestada por mi tutor fue un poco más sencillo.

VI.4. *Aportaciones personales.*

Una de las aportaciones personales fue el hecho de conocer profundamente cómo trabaja el simulador de redes utilizado. En la actualidad el simulador OPNET Modeler está siendo utilizado por muchas universidades en todo el mundo, pero esto no implica que la información de manejo y utilización sea muy elevada, sobretodo respecto a los protocolos ad hoc que nosotros hemos presentado. Por lo que el proceso de aprendizaje no puede considerarse despreciable.

Además uno de los principales problemas que tuvimos fue el hecho de crear grupos, esto como hemos visto anteriormente fue una tarea difícil. Al final investigando en todas la librerías del programa observamos que había un modulo que podía crear agrupaciones de nodos, así que lo probamos y vimos que realizaba la tarea que queríamos.

Otra aportación que hemos introducido en este trabajo es el hecho de otorgar las características correspondientes a los nodos de la red, que la topología inicial y el patrón de movimiento de los nodos sea lo más independiente posible de los datos obtenidos y que el tráfico de datos introducido se adecue al tipo de red que estamos simulando.

Una de las aportaciones más importantes y de la cual se espera sacar más resultados es la arquitectura propuesta. A raíz de esta idea se espera desarrollar una arquitectura óptima para redes WAHSN y una vez desarrollada la idea intentar implementarla sobre alguna red real.

Por último, como fruto del presente trabajo y de otros estudios relacionados con la idea de las redes basadas en grupos, se han publicado en congresos internacionales las referencias [40], [41], [45], [46], [47], [48] y [49] donde los congresos de [48] y [47] son de tipo A y B respectivamente. En revistas internacionales se han publicado [50], [51], [52] y [53] donde las referencias [53] y [52] tienen actualmente un índice JCR de 0.441 y 1.681 respectivamente.

VI.5. *Futuras líneas de trabajo.*

A partir de todos los datos obtenidos en este trabajo y de otros qué no han sido incluidos porque la extensión de la tesina hubiera sido demasiada, están apareciendo diversas líneas de trabajo.

La primera de ellas sería, una vez comprobado que el hecho de utilizar grupos aporta beneficios, desarrollar de manera completa la arquitectura propuesta en el punto cinco.

Actualmente estamos trabajando en el IGIC (Instituto de Gestión Integrada de zonas Costeras) en un proyecto denominado KM3Net. El objetivo principal de este proyecto es la creación del mayor telescopio para la detección de neutrinos. En este proyecto ya existe una parte de comunicación entre sensores, pero se está evaluando la utilización de otro mecanismo basado en grupos para mejorar el rendimiento.

Por otra parte el instituto de investigación está evaluando la creación de una propuesta de proyecto de investigación para la detección de fenómenos subacuáticos, donde se podría aplicar la arquitectura que desarrollaremos, creando así lo que es conocido como UWASN (UnderWater Acoustic Sensor Networks).

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente a través del proyecto incluido en el 6º Programa Marco de la Unión Europea E.U FP6 contract 011937 KM3Net (Design Study for a Deep Sea Facility in the Mediterranean for Neutrino Astronomy and Associated Sciences).

REFERENCIAS

[1] D. Johnson, Y. Hu, D. Maltz. "The Dynamic Source Routing Protocol (DSR) for Mobile Ad-hoc Networks for IPv4". RFC 4728. February, 2007.

[2] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing". RFC 3561. July, 2003.

[3] T. Clausen and P. Jacquet. "Optimized Link State Routing Protocol". RFC 3626. Oct. 2003.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks, in IEEE Communications Magazine, Vol. 40, pp. 102- 114, 2002.

[5] M. Frodigh, P. Johansson, P. Larsson. “Wireless ad hoc networking. The art of networking without a network”. Ericsson Review, No. 4, pp. 248-263, 2000.

[6] S. Kumar, V. S. Raghavan, J. Deng. "Medium Access Control protocols for ad hoc wireless networks: A survey". Ad Hoc Networks. Volume 4, Issue 3, pp. 326-358, May 2006.

[7] E. M. Royer and C.-K. Toh, “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks” IEEE Personal Communications, Vol. 6, No. 2, pp. 46-55, April 1999.

[8] R. Rajaraman, “Topology control and routing in ad hoc networks: a survey”. ACM SIGACT News. Volume 33, Issue 2, Pp. 60-73, June 2002.

[9] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, “Topologically-Aware overlay construction and server selection,” in Proceedings of InfoCom, pp. 1190–1199, New York (USA), June 2002.

[10] Q. Lv, S. Ratnasamy, and S. Shenker, “Can heterogeneity make Gnutella scalable?,” in Proc. of the First Int. Workshop on Peer-to-Peer Systems, pp. 94–103, MIT, Cambridge, MA (USA), 7-8 March 2002.

[11] OPNET Modeler website. Available at, http://www.opnet.com/solutions/network_rd/modeler.html

[12] X Hong, K Xu, M Gerla. "Scalable routing protocols for mobile ad hoc networks". Network, IEEE. Volume 16, Issue 4, Pp. 11-21. Jul/Aug 2002.

[13] José Sabater Alepuz. "Propagation Model For Ad-Hoc & Sensor Networks". Proyecto final de carrera. ETSIT-Universidad Politécnica de Valencia, 2007.

[14] Carlos Domingo Ortiz. "Análisis Y Diseño De Redes De Sensores Inalámbricos 2d Y 3d". Proyecto final de carrera. ETSIT-Universidad Politécnica de Valencia, 2007.

[15] Recurso web del grupo MANET del IETF: http://www.ietf.org/html.charters/manet-charter.html

[16] Carlos Miguel Tavares de Araújo Cesariny Calafate. "Analysis and design of efficient techniques for video transmission in IEEE 802.11 wireless ad hoc networks". Tesis doctoral. Universidad Politécnica de Valencia, 2006.

[17] Luis Girones Quesada. "A Routing Protocol For Manets". Proyecto final de carrera. ETSIT-Universidad Politécnica de Valencia, 2007.

[18] Miguel García Pineda. "Análisis y comparativa del rendimiento de los protocolos de encaminamiento MANET en redes de sensores inalámbricas". Proyecto final de carrera. ETSIT-Universidad Politécnica de Valencia, 2007.

[19] Jaime Lloret Mauri. "Arquitectura de interconexión de redes P2P parcialmente centralizadas". Tesis doctoral. Universidad Politécnica de Valencia, 2006.

[20] K. Woolston, and S. Albin, “The Design of Centralized Networks with Reliability and Availability Constraints”, Computers and Operations Research, Vol. 15, issue 3, pp 207-217, May 1988.

[21] Liu-Sheng Huang, Hong-Li Xu, Yang Wang, Jun-Min Wu and Hong Li, “Coverage and Exposure Paths in Wireless Sensor Networks”, J. of Computer Science and Technology, Vol. 21, issue 4, pp. 490-495, July 2006.

[22] Jaeook Lee and Sun Kang, “Satellite over Satellite (SOS) Network: A Novel Architecture for Satellite Network”, IEEE Infocom 2000, Tel Aviv (Israel), March 26-30, 2000.

[23] A. Ganz, C. M. Krishna, D. Tang, Z.J. Haas, "On optimal design of multitier wireless cellular systems," Communications Magazine, IEEE, vol.35, issue 2, pp.88-93, February 1997.

[24] Chong-Yi Yuan, Wen Zhao, Shi-Kun Zhang and Yu Huang, “A Three-Layer Model for Business Processes — Process Logic, Case Semantics and Workflow Management”, Journal of Computer Science and Technology, Vol. 22, issue 3, pp. 410-425, May 2006.

[25] A. Wierzbicki, R. Strzelecki, D. Swierczewski, and M. Znojek, “Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications”, Second IEEE International Conference on Peer-to-Peer Computing, Linöping (Sweden), 5-7 September 2002.

[26] Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y. Peer-to-Peer Based Multimedia Distribution Service, IEEE Transactions on Multimedia Vol. 6, issue 2, pp. 343-355, April 2004.

[27] L. Hongjun, L. P. Luo and Z. Zhifeng. “A structured hierarchical P2P model based on a rigorous binary tree code algorithm”, Future Generation Computer Systems, Volume 23, Issue 2, pp. 201-208, February 2007.

[28] B. Thallner, H. Moser. “Topology control for fault-tolerant communication in highly dynamic wireless networks”. Proceedings of the 3rd International Workshop on Intelligent Solutions in Embedded Systems. Hamburg University of Technology, Hamburg (Germany), May 20, 2005.

[29] Yu, J.Y., Chong, P.H.J. “A survey of clustering schemes for mobile ad hoc networks”. IEEE Communications Surveys & Tutorials, Vol. 7, Issue: 1, pp: 32- 48, 2005.

[30] M. Jiang, J. Li, Y.C. Tay. “Cluster Based Routing Protocol (CBRP)”. Internet-draf, draft-ietf-manet-cbrp-spec-01.txt, National University of Singapore, 14 August 1999.

[31] Chiang, C.C., Wu, H.K., Liu, W., Gerla, M., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", The Next Millennium, the IEEE SICON, Singapore, National University of Singapore, Kent Ridge (Singapore). April 14-17, 1997.

[32] Zygmunt J. Hass., Marc R. Pearlman, Prince Samar. “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, IETF, Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

[33] Z. J. Haas and M. R. Pearlman, “The zone routing protocol: A hybrid framework for routing in ad hoc networks,” in Ad Hoc Networks, C. E. Perkins, Ed. Reading, MA: Addison-Wesley, 2000.

[34] Mirco Musolesi, Cecilia Mascolo. “A Community Based Mobility Model for Ad Hoc Network Research”, Second International Workshop on Multi-hop Ad hoc Networks: from theory to reality REALMAN 2006, Florence (Italy), 26 May 2006.

[35] P. F. Tsuchiya, “The Landmark Hierarchy: a new hierarchy for routing in very large networks,” In Computer Communication Review, vol.18, issue.4, pp. 35-42, August 1988.

[36] Y. Rekhter, T. Li, “A Border Gateway Protocol 4 (BGP-4)”, RFC 1771, March 1995.

[37] G. Pei, M. Gerla and X. Hong, “LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility”, in Proceedings of IEEE/ACM MobiHOC, pp. 11-18, Boston, MA (USA), August 2000.

[38] Srdjan Krco, “Health Care Sensor Networks – Architecture and Protocols”, Ad Hoc & Sensor Wireless Networks, Vol. 1, issue 1-2, pp. 1-25, 2005.

[39] E. M. Sozer, M. Stojanovic, and J. G. Proakis, “Underwater Acoustic Networks”, IEEE Journal of Oceanic Engineering, Vol. 25, No. 1, pp. 72-83, January 2000.

[40] J. Lloret, M. Garcia, and J. Tomas, “Improving Mobile and Ad-hoc Networks performance using Group-Based Topologies” in IFIP International Federation for Information Processing, Volume 264; Wireless Sensor and Actor Networks II; Ali Miri; (Boston: Springer), pp. 209–220, 2008.

[41] J. Lloret, M. García, F. Boronat, and J. Tomás, “MANET Protocols Performance in Group-based Networks” in IFIP International Federation for Information Processing, Volume 284; Wireless and Mobile Networking; Zoubir Mammeri; (Boston: Springer), pp. 161–172, 2008.

[42] P. Gober, A. Ziviani, P. Todorova, M. Dias de Amorim, P. Hunerberg, S. Fdida, “Topology Control and Localization in Wireless Ad Hoc and Sensor Networks”, Ad Hoc & Sensor Wireless Networks, Vol. 1, issue 4. 2005.

[43] J. M. McQuillan, I. Richer, E. C. Rosen, “The New Routing Algorithm for the ARPANET”. IEEE Transactions on Communications, Volume: 28, Issue 5, pp.711-719. May 1980.

[44] Y. K. Dalal, R. M. Metcalfe, “Reverse path forwarding of broadcast packets”, Communications of the ACM. Vol. 21, issue 12, pp: 1040-1048. December 1978.

[45] J. Lloret, M. García and J. Tomás, “A Group-Based Architecture for Wireless Sensor Networks”. International Conference on Networking and Services (ICNS'07), Athens (Greece), June 19-25, 2007.

[46] M. Garcia, D. Bri, F. Boronat and J. Lloret, “A new neighbor selection strategy for group-based wireless sensor networks”. The Fourth International Conference on Networking and Services (ICNS 2008), Gosier, Guadeloupe, March 16-21, 2008.

[47] J. Lloret, M. Garcia, F. Boronat and J. Tomás, "A Group-Based Protocol for Large Wireless AD-HOC and Sensor Networks". IEEE Network Operations and Management Symposium Workshop, NOMS 2008, Salvador de Bahia, Brasil, 7-11 de April de 2008.

[48] J. Lloret, M. Garcia, D. Bri and J. R. Diaz, "A Group-based Content Delivery Network". ACM/IEEE International Symposium on High Performance Distributed Computing 2008 (HPDC 2008), Boston, MA (USA), 23-27 de June de 2008.

[49] M. Garcia, H. Coll, D. Bri and J. Lloret, "Using MANET protocols in Wireless Sensor and Actor Networks". The Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008), Cap Esterel, Costa Azul (Francia), 25-31 de August de 2008.

[50] J. Lloret, M. Garcia, F. Boronat and J. Tomas, "Group-based Self-Organization Grid Architecture". Lecture Notes in Computer Science, Vol. 4459, pp. 590-602, May 2007.

[51] F. Boronat Seguí, J. C. Guerri Cebollada, J. Lloret Mauri, M.García Pineda, "Sincronización de grupo multimedia basada en protocolos estándar". IEEE Latin America Transactions, Vol. 5, issue 6, pp. 457-464, October 2007.

[52] F. Boronat, J. Lloret and M. Garcia, "Multimedia Group and Inter-Stream Synchronization Techniques: A Comparative Study". Information Systems, doi:10.1016/j.is.2008.05.001, Available online 24 May 2008.

[53] J. Lloret, M. Garcia, J. Tomás and F. Boronat, "GBP-WAHSN: A Group-Based Protocol for Large Wireless Ad Hoc and Sensor Networks". Journal of Computer Science and Technology, Vol. 23, issue 3, pp. 461- 480, May 2008.

ANEXO

A continuación presentamos los artículos que han nacido a raíz de esta tesina y de otros estudios relacionados con la misma. Los artículos que se muestran a continuación están ordenados por fecha de publicación. Seguidamente presentamos una tabla resumen para aportar mayor claridad al lector.

AUTORES (p.o. de firma): *Jaime Lloret, Miguel Garcia, Fernando Boronat y Jesús Tomás*
TÍTULO: *Group-based Self-Organization Grid Architecture*

EDITORIAL: *Springer-Verlag Berlin Heidelberg*
REF. REVISTA/LIBRO: *Lecture Notes in Computer Science.* (ISBN, ISSN,...): *ISSN: 0302-9743* CLAVE: A
VOLUMEN: *4459* PÁGINAS: *590-602* AÑO: *Mayo 2007*

AUTORES: *Jaime Lloret, Miguel Garcia y Jesus Tomas*
TÍTULO: *A Group-Based Architecture for Wireless Sensor Networks*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *International Conference on Networking and Services (ICNS'07)*
PUBLICACIÓN: *IEEE Computer Society* ISBN: *978-0-7695-2858-9*
LUGAR DE CELEBRACIÓN: *Atenas (Grecia)*
AÑO: *19-25 de Junio de 2007*

AUTORES: *Fernando Boronat Seguí, Juan Carlos Guerri Cebollada, Jaime Lloret Mauri, Miguel García Pineda*
TÍTULO: *Sincronización de grupo multimedia basada en protocolos estándar*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *VI JORNADAS DE INGENIERÍA TELEMÁTICA (JITEL 2007)*
PUBLICACIÓN: *Libro de Actas del Congreso* ISBN: *978-84-690-6670-6*
LUGAR DE CELEBRACIÓN: *Málaga (España)*
AÑO: *17-19 de Septiembre de 2007*

AUTORES: *Fernando Boronat Seguí, Juan Carlos Guerri Cebollada, Jaime Lloret Mauri, Miguel García Pineda*
TÍTULO: *Sincronización de grupo multimedia basada en protocolos estándar*

EDITORIAL: *IEEE*
REF. REVISTA/LIBRO: *IEEE Latin America Transactions* (ISBN, ISSN,...): *ISSN: 1548-0992* CLAVE: A
VOLUMEN: *5*, Nº: *6* PÁGINAS: *457-464* AÑO: *Octubre 2007*

AUTORES: *Miguel Garcia, Diana Bri, Fernando Boronat y Jaime Lloret*
TÍTULO: *A new neighbor selection strategy for group-based wireless sensor networks*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *The Fourth International Conference on Networking and Services (ICNS 2008)*
PUBLICACIÓN: *IEEE Computer Society* ISBN: *0-7695-3094-X*
LUGAR DE CELEBRACIÓN: *Gosier, Guadeloupe*
AÑO: *16-21 de Marzo de 2008*

AUTORES: *Jaime Lloret, Miguel Garcia, Fernando Boronat y Jesus Tomás*
TÍTULO: *A Group-Based Protocol for Large Wireless AD-HOC and Sensor Networks*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *IEEE Network Operations and Management Symposium Workshop, NOMS 2008*
Listada en el "Computing Research and Education (CORE)"como B y en el CiteSeer
PUBLICACIÓN: *IEEE Computer Society* ISBN: *978-1-4244-2067-4*
LUGAR DE CELEBRACIÓN: *Salvador, Bahia, Brasil*
AÑO: *7-11 de Abril de 2008*

AUTORES (p.o. de firma): *Fernando Boronat, Jaime Lloret y Miguel Garcia*
TÍTULO: *Multimedia Group and Inter-Stream Synchronization Techniques: A Comparative Study*

EDITORIAL: *Elsevier B. V.*
REF. REVISTA/LIBRO: *Information Systems.* CLAVE: A
(ISBN, ISSN,...): *ISSN: 0306-4379* ***Available online 24 May 2008***
VOL: Issue: PÁGINAS: 42 AÑO: *2008* **J.C.R.: 1,681**
doi:10.1016/j.is.2008.05.001

AUTORES (p.o. de firma): *Jaime Lloret, Miguel Garcia, Jesus Tomás y Fernando Boronat*
TÍTULO: *GBP-WAHSN: A Group-Based Protocol for Large Wireless Ad Hoc and Sensor Networks*

EDITORIAL: *Springer-Verlag Berlin Heidelberg*
REF. REVISTA/LIBRO: *Journal of Computer Science and Technology.* CLAVE: A
(ISBN, ISSN,...): *ISSN: 1000-9000*
VOL: 23 Issue: 3 PÁGINAS: *461-480* AÑO: *Mayo 2008* **J.C.R.: 0,441**

AUTORES: *Jaime Lloret, Miguel Garcia, Diana Bri y Juan R. Diaz*
TÍTULO: *A Group-based Content Delivery Network*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *ACM/IEEE International Symposium on High Performance Distributed Computing 2008 (HPDC 2008)*
Listada en el "Computing Research and Education (CORE)" como HPDC como A
PUBLICACIÓN: *ACM* ISBN*: 978-1-60558-155-2*
LUGAR DE CELEBRACIÓN: *Boston, MA (USA)*
AÑO: *23-27 de Junio de 2008*

AUTORES: *Jaime Lloret, Miguel García y Jesús Tomás*
TÍTULO: *Improving Mobile and Ad-Hoc Networks Performance using Group-based topologies*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *Wireless Sensors and Actor Networks 2008 (WSAN 2008)*
PUBLICACIÓN: *IFIP Series, Springer Boston* ISBN: *978-0-387-09440-3*
LUGAR DE CELEBRACIÓN: *Ottawa (Canada)*
AÑO: *14-15 de Julio de 2008*

AUTORES: *Miguel Garcia, Hugo Coll, Diana Bri, Jaime Lloret*
TÍTULO: *Using MANET protocols in Wireless Sensor and Actor Networks*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *The Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008)*
PUBLICACIÓN: *IEEE Computer Society* ISBN: *978-0-7695-3330-8*
LUGAR DE CELEBRACIÓN: *Cap Esterel, Costa Azul (Francia)*
AÑO: *25-31 de Agosto de 2008*

AUTORES: *Jaime Lloret, Miguel García, Fernando Boronat, Jesus Tomás*
TÍTULO: *MANET protocols performance in group-based Networks*

TIPO DE PARTICIPACIÓN: *Ponencia*
CONGRESO: *10th IFIP International Conference on Mobile and Wireless Communications Networks*
PUBLICACIÓN: *IFIP Series, Springer Boston* ISBN: *978-0-387-84838-9*
LUGAR DE CELEBRACIÓN: *Toulouse (Francia)*
AÑO: *29-30 de Septiembre de 2008*

Group-Based Self-organization Grid Architecture

Jaime Lloret[1], Miguel Garcia[2], Fernando Boronat[3], and Jesus Tomas[4]

Communications Department, Polytechnic University of Valencia
Camino Vera s/n, 46022 Valencia, Spain
{jlloret,fboronat,jtomas}@dcom.upv.es, migarpi@teleco.upv.es

Abstract. Many grid architectures have been developed since the first proto-grid systems in the early 70's, but there are not so many based on groups using an efficient node neighbor selection. This paper proposes a grid architecture based on groups. The architecture organizes logical connections between nodes from different groups of nodes allowing sharing resources, data or computing time between groups. Connections are used to find and share available resources from other groups and they are established based on node's available capacity. Suitable nodes have higher roles in the architecture and their function is to organize connections based on a node selection process. Nodes' logical connections topology changes depending on some dynamic parameters. The architecture is scalable and fault-tolerant. We describe the protocol, its management and real measurements. It could be used as an intergrid protocol.

Keywords: Grid architecture, group-based logical network, neighbor selection, peer-to-peer network, intergrid protocol.

1 Introduction

Grid computing provides always-online computer services to users. It reduces significantly computation time on complex problems. A grid is a system that is concerned with the integration, virtualization and management of services and resources in a distributed and heterogeneous environment. It supports collections of users and resources across traditional administrative and organizational domains that are able to manage and run some processes to carry out an objective [1]. It enables the integrated and collaborative use of high-end computers, networks, databases and scientific instruments, owned and managed by multiple organizations, giving coordinated resource-sharing and problem-solving capabilities to its users.

There are many projects around the world working on developing grids for different purposes at different scales from the academic research communities, from the industry and from government-sponsored infrastructure projects. Grid computing was primarily used to support scientific research into large problems concerning weather, astronomy, and medicine, but the number of potential applications seems to grow every year, because of the increasing corporate interest in turning the technology into business. New applications are based on protocols developed for specific purposes such as the parallel filesystem [2], data storage systems [3], data replication and retrieval systems [4] and data processing systems [5].

The paper is structured as follows. Section 2 examines some Grids architectures, works related with our proposal such as neighbor selection, hierarchical architectures

C. Cérin and K.-C. Li (Eds.): GPC 2007, LNCS 4459, pp. 590 – 602, 2007.

and architectures based on groups, and explains our motivation. There is a description of our architecture proposal in section 3. Analytical model for some types of topologies of nodes used in our architecture and our analysis is explained in section 4. The protocol operation, recovery algorithms and designed messages are shown in section 5. Section 6 shows the performance operation when the architecture is running. Finally, section 7 gives our conclusions and future works.

2 Previous Works and Motivation

In this section we will relate several known grids architectures, we will describe several strategies to establish connections between nodes and, finally, we will explain several works where nodes are divided into groups. It will give the lecturer the state of the art related with our architecture, because it establishes connections between the more suitable nodes from different groups.

Condor Project was born to take advantage of the idle time of the computers in the network. It is a high-throughput distributed batch computing system. Condor is based on a centralized architecture where users submit their jobs, and it chooses when and where to run them based upon a policy, monitors their progress, and finally informs the user upon completion. The NorduGrid project's primary goal is to meet the requirements of production tasks of LHC (Large Hydron Collider) experiments. The NorduGrid topology is decentralized, avoiding a single point of failure. It is a lightweight, non-invasive and dynamic one, while robust and scalable, capable of meeting most challenging tasks of High Energy Physics. These infrastructures use a software platform to organize and run the jobs. Although Globus Toolkit™ is one of the most used, there are others such as Netsolve, Nimrod and AliEn. These production environments implement virtual topologies in distributed ways were nodes establish connections, to become neighbors, as needed to coordinate resources and services.

Throughout the years different types of strategies for neighbors' selection have been developed. Simon et al., in [6], proposed a genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks, which enhances the decision process performed at the tracker for transfer coordination increasing content availability to the clients from their immediate neighbors. There are proposals where nodes' connections are based on the underlying network, such as Plethora [7] or on their geographic location such as the one described by K. Liu et al. in [8]. Others systems, such as the one presented by X. Zhichen in [9], locate nodes in the topology taking into account that are possibly close to a given node, and then perform RTT measurements to identify the actual closest node.

There are several works in the literature where nodes are divided into groups and connections are established between nodes from different groups, but all of them are developed to solve specific issues. To the extent of our knowledge, there is not any previous interconnection system to structure connections between groups of nodes like the one that will be presented in this paper. A. Wierzbicki et al. presented Rhubarb [10]. It organizes nodes in a virtual network, allowing connections across firewalls/NAT, and efficient broadcasting. The system uses a proxy coordinator. When a node from outside the network wishes to communicate with a node that is inside, it sends a connection request to the proxy coordinator, who forwards the request to the node inside the network. Rhubarb uses a three-level hierarchy of groups, may be sufficient to support a million nodes, but when there are several millions of nodes in the network it could not be enough, so it suffers from scalability problems. On the other hand, all nodes need to know the IP of the proxy coordinator nodes to establish connections with nodes from other virtual networks. Z. Xiang et al. presented a Peer-to-Peer Based Multimedia Distribution Service [11]. It proposes a topology-aware overlay in which nearby peers self-organize into application groups. End hosts within the same group have similar network conditions and can easily collaborate with each other to achieve QoS awareness. When a node in this architecture wants to communicate with a node from other group, the information is routed through several groups until it arrives to the destination. There are some hierarchical architectures were nodes are structured hierarchically and parts of the tree are grouped into groups such as the one presented by Liu Hongjun et al. in [12]. The information has to be routed through the hierarchy to achieve nodes from other groups, so all layers of the hierarchy could be overloaded in case of having many data to be transferred. On the other hand, in case of many groups, the hierarchical structure could become unstructured because there could be many connections establishments between nodes from different groups placed on different layers of the hierarchy.

Grids architectures could be deployed different according to the necessities of the final purpose. Let's suppose we need to organize the grid into groups in order to process parts of an application in parallel, but in certain moments, nodes from a group need some resources, data or computation time from other groups. All architectures previously shown don't solve that problem efficiently, because in the case of centralized architectures, such as Condor project, the server will have many logical connections at the same time to distribute jobs, so it will need many resources. On the other hand, there is a central point of failure and a bottleneck. In the case of fully distributed architectures, the control system use to be very difficult to be implemented and it needs much time to process tasks because of the time needed to reach far nodes. It decreases the performance of the whole system. To address this problem, we propose an architecture based on groups where nodes work in their group as in a regular grid, but they can reach all other groups, if needed, in one hop, diminishing the time to reach resources, data or computing from other groups enhancing the performance of the whole system.

3 Architecture Outline

We propose to split the grid network in groups of nodes. Nodes can reach all nodes in their group to coordinate and sharing resources and services and some of them will have logical connections (from now we will call just connections) with nodes from other groups based on some parameters defined later. A node will collaborate with nodes from its group as a small network and when a node (or the group of nodes) needs data, resources or computing time from another group, one of them requests it to the other group. The reply is sent to the requesting node, and in case of data, it can share it acting as a cache for its group.

Nodes in the proposed architecture could be a regular node or could have one or several of the following roles (a node could run all them simultaneously, depending on its functionality in the group): (i) Distribution role node (DN): A DN will have a connection with one node (becoming adjacent) from each other groups as a hub-and-spoke. The number of connections to other groups can be limited by several parameters described later. Connections are used to send searches for resources, data or computing time between groups. (ii) Area controller role node (AC): ACs organize DNs in zones to have an scalable architecture. They are able to reach a GC in its group and to choose the best DN in their area. (iii) Group controller role node (GC): It could be one or several in each group, depending on the number of DNs in the group. GCs have connections with GCs from other groups. A GC has AC functionalities too, so it has connections with ACs from its group. Both ACs and GCs have DN functionalities. GC organizes nodes in its group and adjacencies between DNs from different groups. From now, we will not consider regular nodes because the proposed architecture works without these leaf nodes, but regular nodes will know how to reach a DN in its grid (it could be announced as a service in the grid protocol).

Figure 1 show a topology example. The network topology of each group could be different, but all nodes in the topology run the same application layer protocol.

When a node joins a group it acquires a unique node identifier (*nodeID*). The first node in a group will have *nodeID*=0x01, and it will assign *nodeIDs* sequentially to new ones. All nodes in a group have the same *groupID*. We define δ as the node promotion parameter. It depends on node's bandwidth and its *nodeID*. It is used to know which node is the best one to have higher role. Nodes with higher bandwidth and older (lower *nodeID*) are preferred to promote. Every β DNs, the DN with higher δ in the group will start AC role and it will create a new area. Every α ACs, the AC with higher δ will start GC role. α and β values depend on the number of nodes in the group and the network topology of the group and they will be discussed in the analytical model section (next section). We define λ as the node capacity. It determines the best node to have an adjacency with. It depends on node's bandwidth, its number of available connections, its maximum number of connections and its % of available load.

We have chosen Short Path First (SPF) algorithm to route information between GCs and between ACs using a two-level SPF-Based System such as the one described by some authors of this paper in [13]. It is fast and allows sending fast searches to find DN adjacencies, but it can be changed by other routing protocol depending on the networks' features. GCs route information using *groupID* parameter and ACs route information using *nodeID* parameter. Link cost (C) between nodes is based on node's capacity. The more the node's capacity is, the lower its cost is. Every GC or AC runs SPF algorithm locally and selects the best path to a destination node based on a metric. The metric is based on the number of hops to a destination and the link cost of those nodes involved in the path. Experiments given in [14] show that a database having 10^4 external updates from other GCs will consume 640 Kbytes of memory. Table 1 summarizes all parameters described. Expressions proposed in table 1 for δ, λ, C and Metric are based on proves and simulations used for Multimedia Networks [15]. We estimate they fit our architecture proposal requirements.

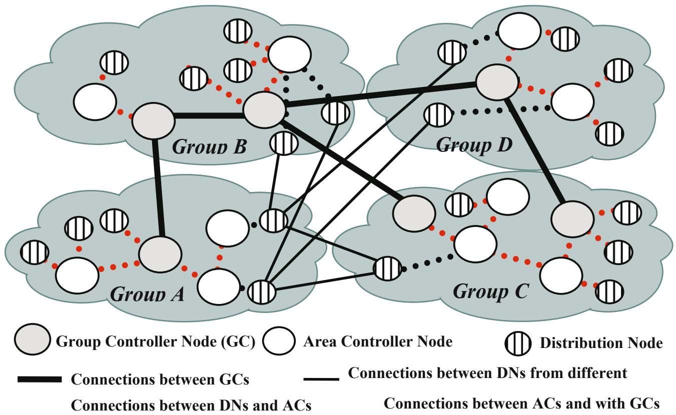


Fig. 1. Architecture organization

Table 1. Parameters summary

Description	Symbol	Expression
Node identifier	*nodeID*	-
Group identifier	*groupID*	-
Parameter to promote a new AC	β	-
Parameter to promote a new GC	α	-
Maximum number of Connections	*Max_Con*	-
Available number of Connections	*Available_Con*	-
Constants used to adjust the weigh of some parameters in the expressions	K_1, K_2, K_3, K_4	-
Node promotion parameter	δ	$\delta = (BW_{up} + BW_{down}) \cdot K_1 + (32 - \log_2(nodeID)) \cdot K_2$
Node capacity	λ	$\lambda = \dfrac{\mathrm{int}\left[\dfrac{(BW_{up} + BW_{down})}{256} + 1\right] \cdot Available\_Con \cdot (100 - load) + K_3}{Max\_Con}$
Link cost	*C*	$C = \dfrac{K_4}{\lambda}$
Metric for node *j*	*Metric (j)*	$Metric(j) = \sum_{i=1}^{n} C_i$

4 Analytical Model and Analysis

In this section we are going to describe the architecture analytically in terms of group of nodes and we will suppose several types of logical topologies for all groups. It allows us to know how many connections will be in our proposal using each one of the logical topologies implemented to validate our model.

Given $G = (V, \lambda, E)$ a network of nodes, where V is a set of DNs (ACs and GCs are DNs too), λ is a set of capacities ($\lambda(i)$ is the i-DN capacity and $\lambda(i) \neq 0$ $\forall$ i-DN) and E is a set of connections between DNs. Let k be a finite number of disjoint subsets of V. V_k is the subset k and $V = \cup (V_k)$. Given a DN_{ki} (i-th DN from the k subset), it will not have any connection with DNs from the same subset ($e_{ki\text{-}kj}=0$ $\forall$ V_k). Every DNv_{ki} has a connection with one DN_{ri} from other subset $(r \neq k)$. Let's suppose $n=|V|$ and k the number of subsets of V, then we obtain equation 1.

$$n = \sum_{i=1}^{k} | V_k | \tag{1}$$

Every V_k has regular nodes and DNs (GCs and ACs are DNs too). So, nodes of every group are the sum of all of them. Now we can describe the whole network as a sum of regular nodes and DNs by expression 2.

$$n = \sum_{i=1}^{k} | (n_{regular} + n_{DN})_k | = \sum_{i=1}^{k} (|n_{regular}|)_k + \sum_{i=1}^{k} (|n_{DN}|)_k \tag{2}$$

Regular nodes will be the interior nodes of the topology and DNs will be edge nodes. There are several known laws where the number of interior nodes is related to the edge nodes.

M. Faloutsos et al. show in [16] that many networks could be modelled following several mathematical models. It also shows that the power law fit the real data in correlation coefficients of 96% in Internet. Based on power law we can find Zipf's law, which states that few nodes have many connections while there are many nodes with few connections. B. A. Huberman and L. A. Adamic in [17] proposed the Zipf's law for Internet and Z. Ge et al. proposed Zipf's law for Gnutella and Napster networks in [18]. The mathematical expression for power law that relates edge nodes with interior nodes, and adapted to our case, is given in expression 3.

$$n_{DN} = \frac{1}{2(R+1)}(1 - \frac{1}{n_{regular}^{R+1}})n_{regular} \tag{3}$$

Where n_{DN} is the number of edge nodes, $n_{regular}$ is the number of interior nodes and R varies as a function of the network where it is applied. In the case of Internet it has been varying along the years having -0.81, -0.82 and -0.74 values.

György Hermann introduced another mathematical model in [19]. It proposes, using D. J. Watts and H. S. Strogatz networks model [20], where network connections are established based on efficiency, stability and safety properties. Expression 4 gives their proposed relationship.

$$c \cdot n_{regular} \leq n_{DN} \leq c \cdot n_{regular} \cdot \ln(n_{regular}) \tag{4}$$

Where n_{DN} is the number of edge nodes, $n_{regular}$ is the number of interior nodes and c is a constant which value depends on the network model

In [13], the same authors of this paper propose different relationship between regular nodes and distribution nodes for partially centralized P2P networks. If we are talking about an hybrid P2P network, the number of edge nodes could be equal to the number of regular nodes, but in case of a superpeer P2P network, it is needed a distribution node every 96 regular nodes. Expression 5 summarized these values.

$$n_{DN} = \begin{cases} n_{regular} & \text{in case of a hybrid P2P network} \\ \frac{n_{regular}}{96} & \text{in case of a superpeer P2P network} \end{cases} \tag{5}$$

Figure 2 shows the number of nodes in a group as a function of the number of regular nodes in a group of the proposed architecture. The hybrid P2P network is the same case of minimum value of the Hermann model (Hermann_min).

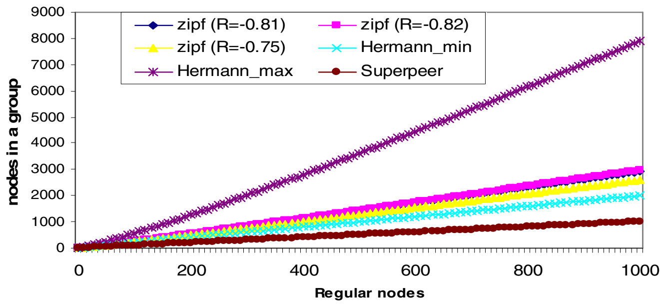


Fig. 2. Number of nodes in a group as a function of the regular nodes

Using Herman maximum value (Hermann_max), we need many nodes in the group, so there will be many DNs. On the other hand, the one that will need less DNs in the group will be topologies such as the superpeer P2P network

5 Protocol Operation

First node in the network starts with *groupID*=0x01 and *nodeID*=0x01 and has all roles in its group. Next new nodes in that group enter as DNs and will acquire roles as a function of their δ. In order to join new groups to the architecture, the GC of the new group must to send a "GG discovery" message, with its *groupID*, to GCs from other groups known in advance or by bootstrapping [21] (a *groupID* value of 0xFF indicates the architecture must assign next available *groupID* value, and if the new GC has a *groupID* value that is used, it will be invited to change the *groupID* indicating next *groupID* available). If there is not any reply in a certain period of time, it will begin the process again. GCs from other groups reply this message with their *networkID* and their λ parameter in the "GC discovery ACK" message. It chooses GCs with higher λ and sends them a "GC connect" message. Then, they reply with a "GC welcome" message indicating that it has joined the architecture. After that, it sends them its neighbor list using "GCDB" message. Its neighbors add this entry to their topological database and recalculate routes using SPF algorithm. When they finish, they will send their database to the new GC to build its database. Next database messages will be updates only. Finally, it will send them "keepalive GC" messages periodically to indicate that it is still alive. If a GC does not receive a "GC keepalive" message from a neighbor for a holdtime, it will erase this entry from its database.

New joining nodes in a group will be DNs. A DN sends a "D discovery" message to ACs previously known or by bootstrapping. Only ACs of its group will reply using "D discovery ACK" messages with their *groupID* and λ. DN will choose the AC with higher λ and it will send it a "D connect" message. AC will reply a "Welcome D" message with assigned *nodeID*. Then, it will add DN's entry to its access table (the owner is the AC of an area and it is formed by all DNs in that area). Finally, DN will send it "Keepalive D" messages periodically. If the AC does not receive a "Keepalive D" message from a DN for a holdtime, it will erase this entry from its table. Next, DN has to establish an adjacency with DNs from other groups, so it will send a "DDB request" message to the AC in its zone. This message contains sender's *groupID*, sender's *nodeID* and its network layer address and the destination groupID (0x00 in case of "all groups"). Then, AC routes it to the GC in its group. GC will send this request to all GCs from other groups in its distribution table (GCs' distribution table is formed by all GCs the owner can reach). When a GC receives this message from other group, it will send a "Find DN" message to ACs in its group in order to find the DN with highest λ in the group. Every request has a unique sequence number to avoid route loops in the group. ACs will reply with their 2 DNs with highest λ using the message "Found DN". GC waits replies for a certain period of time. It chooses 2 highest λ DNs and sends them a "Elected DN" message. The highest one will be the preferred; the second one will act as a backup. This message contains the *nodeID* and the requesting DN's network layer address. When these DNs receive that message, they will send a "DD connect" message to connect with the DN from the other group. Next, they send a "D elected ACK" message to the GC in its group to indicate a connection has established with other group DN. If GC does not receive this message for a hold time, it will send a new message to the next DN with highest λ. This process will be repeated until GC receives both confirmations. When the requesting DN from other group receives these connection messages, it will add DN with highest λ as its first neighbor and the second one as the backup. Then, it replies these connection messages to acknowledge the connection using the "DD welcome" message. If the requesting DN does not receive any connection from other DN for a holdtime, it will send a requesting message again. Finally, both DN will send "keepalive DD" messages periodically. If a DN does not receive a "keepalive DD" message from the other DN for a holdtime, it will erase this entry from its DN's distribution table (it is formed by all neighbor DNs from other groups).

When a GC receives a new groupID in a "GC connect" or in a "GCDB" message, it will send a "New group" message to all ACs in its group with a sequence number to avoid route loops. Then, ACs will forward this message to all DNs in their zone. Subsequently, DNs will begin the process to request DNs from the new group.

When a GC sees there are β more ACs in its group, it will send a "GC conversion" message to the AC with highest δ in its AC distribution table (ACs' distribution table is formed by all ACs in the group). Highest δ AC will send a "change level" to its neighbors to inform them it has changed its level and it will begin the process of authenticating with other GCs.

When the oldest GC sees there are β more DNs in its group, it will send an "AC request" message to all ACs to request a new AC. All ACs will reply an "AC reply" message with the *nodeIDs* of the first and the second DNs with highest δ in its group. GC will process all replies and will choose 2 DNs with highest δ from the whole group. Then, it will send an "AC conversion" message to the first DN with highest δ. This message will be routed to the chosen DN. This DN will become an AC and will send an "AC disconnection" message to its AC. If GC does not receive changes in ACs' distribution table for a hold time, it will send a new "AC request" message to the second DN with highest δ. If this time it fails again, it will begin the process, but avoiding those DNs. New ACs must authenticate with ACs in their group. It can establish its first connections with any AC known in advance or by bootstrapping [21]. First, it sends an "AC discovery" message with its *groupID*. Only ACs with the same *groupID* will reply with their λ. New AC will wait for a hold time and will choose

ACs with highest λ. If there is no reply, new AC will send an "AC discovery" message again. Then, new AC will send an "AC connect" message to the chosen ACs. They will reply with a "Welcome AC" message indicating it is connected to the architecture and they will become its neighbors. New AC will send its neighbor list using "AC neighbors" message to all of them to update their AC distribution database and all of them will recalculate new routes using SPF algorithm and the metric aforementioned. Then, they send their database to the new AC using "ACDB" in order to build its ACs' distribution database. Next times it will only receive updates. New AC will send "AC keepalive" messages to its neighbors periodically. If it is not received from a neighbor for a holdtime, it will erase this entry from its database.

5.1 Recovery Algorithms

Every GC sends its backup information to the highest δ AC in the group periodically. When a GC leaves the architecture voluntarily, it will send a "Failed GC" message to the highest δ AC announcing it. The highest δ AC becomes a GC and acknowledges with a "Failed GC ACK" message. Then, GC leaves the architecture sending a "GC disconnect" message to its neighbors. If that GC does not receive the acknowledgement, it will begin the process with the second highest δ AC. Next, new GC sends a "Change level" message to its neighbors to advertise it has changed its level. It will try to have the same neighbors as the old one using the backup data. Then, it will begin its functionalities as a new GC. When a GC fails, it will be detected by its AC neighbors because the lack of "AC keepalive" messages for a holdtime. First AC detects this failure, updates its ACs' database and propagates it through the group using "ACDB" messages. When the highest δ AC receives this update, it will use the backup information and it will become GC.

Every AC has a table with all DNs in its area and information related with its AC neighbor closest to the GC. They will use this table to know their δ and λ. DN with highest δ will be the AC backup DN and it will receive AC backup data from its AC by incremental updates using "Backup AC" messages. This information is used in case of AC failure. AC sends "AC keepalive" messages to the backup DN periodically. When an AC leaves the architecture, it will send a "Failed AC" message to its closest GC with information about its backup DN. The GC will reply it with the "Failed O1 ACK" message, and then, AC will send an "AC disconnect" message to its neighbors and it will leave the architecture. Next, GC, using the received backup data, chooses the highest δ DN in the group (as it has been explained before) and sends it an "AC conversion" message. New AC will send a "DN disconnection" message to its AC, and then, it will connect with the backup DN to have the backup data and become an AC. Then, new AC sends a "Keepalive D" message to all DNs in its zone. If the GC does not receive changes for a hold time, it will send a new request message to the second DN with highest δ. If the backup DN does not receive this message for a hold time, it will become the new AC. When an AC fails, backup DN can check it because the lack of "keepalive D" messages for a holdtime. If it happens, backup DN sends a "Failed AC" message to the failed AC neighbor. It will be the helper AC to help the failed AC substitution. Helper AC will forward the "Failed AC" message to its closest GC to request a new AC. Then, the process will begin as it has been explained before.

When a DN leaves the architecture voluntarily, it will send a "DN disconnect" message to the AC in its zone and to all its adjacent DNs from other groups. They will delete this entry from its DN's distribution database and adjacent DNs will substitute it with a new DN for that group as explained before. When a DN fails down, AC and adjacent DNs will check it because they do not receive a "keepalive D" message for a hold time. Then, AC will delete this entry from its access table and adjacent DNs will delete this entry from its DNs' distribution database and they will request a new DN.

5.2 Protocol Messages

We have designed and developed 46 messages for the architecture operation. We have considered that *networkID*, *nodeID*, λ and δ parameters use 32 bits, so we can classify them in 40 fixed size messages and 6 messages which size depends on the number of neighbors, the size of the topological database or the backup information. Longer messages are the ones that contain the topological database and the backup information. First time, both messages send the whole information, next times only updates are sent.

6 Performance Evaluation

To evaluate the performance of our proposal under real constraints, we have developed a desktop application using Java programming to run and test the proposed architecture and its protocol. It allows the node to run DN, AC and GC roles, as it is described previously, to work the architecture properly. The application let us choose the group connected to and we can vary some parameters such as k_1, k_2, k_3, Max_Con, upstream and downstream bandwidth, keepalive time, timers and so on.

6.1 Testbed

We have used 42 computers (AMD Athlon™ XP 1700+, 1.47 GHz, 480 MB RAM) with Windows XP Professional Operative System. They were connected to several Cisco Catalyst 2950T-24 Switches over 100BaseT links. The implemented scenario has 3 groups interconnected. All these groups have only one GC (which is also an AC). First group has 12 DNs, second group has 13 DNs and the third group has 17 DNs. In order to take measurements from the scenario, we have connected every group to a switch and all Switches were connected to a switch as a star topology. GCs are connected physically to the central switch, although they pertain to their group. One port of the central switch was configured in a monitor mode (receives the same frames as all other ports), to be able to capture data using a sniffer application. We began to take measurements before we started the GC from the first group, 10 seconds later we started the GC from the second group, 10 seconds later we started the GC from the third group, 10 seconds later we began to start all DNs from the first group, 10 seconds later, we started all DNSs from the second group and finally, 10 seconds later, all DNs from the third group.

6.2 Measurement Results

We have used the testbed in 2 cases with different values for keepalive time (20 vs 30 sec.) and timer (4 vs 10 sec.) to evaluate the performance of the system.

Figure 7 a) shows the bandwidth consumed in the testbed for the first case. The number of Bps (Bytes per sec.) oscillates from 4,000 to 8,000 Bps when the network has converged. Peaks because of keepalive messages are not so significant in this case. Figure 7 b) shows the number of messages per sec. in the network when the architecture is running using values of the first case. There are peaks every 20 sec. starting from a 70 sec. approximately because discovery messages and keepalive messages (every 20 sec.), between DNs and the GC, are added. Figure 7 c) shows the number of broadcasts per sec. in the scenario for first case parameter values. The highest peak appears around 70 sec. (when DNs from the third group were started).

Figure 8 a) shows the bandwidth consumed in the network when the architecture is running using values of the second case. The number of Bps oscillates from 2,000 to 8,000 Bps when the network has converged (the number of octets minimum is lower than the first case). Figure 8 b) shows the number of messages per sec. in the scenario for first case parameter values. There are fewer messages per sec. than in the first case and the minimum peaks are lower. Figure 8 c) shows the number of broadcasts per sec. in the testbed for the second case. When the network has converged, there is an average between 2 a 4 of broadcasts per sec. (less than in the first case).

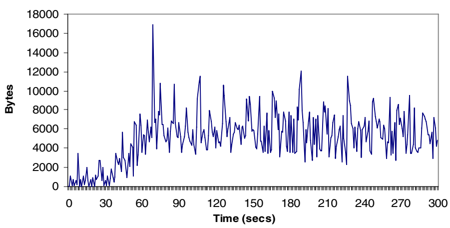


Fig. 7 a). 1st prove bandwidth utilization

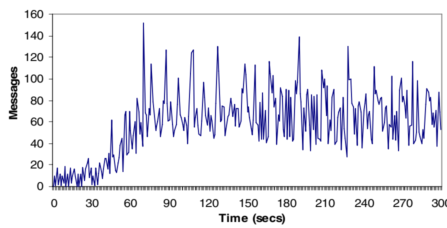


Fig. 7 b). 1st prove number of messages

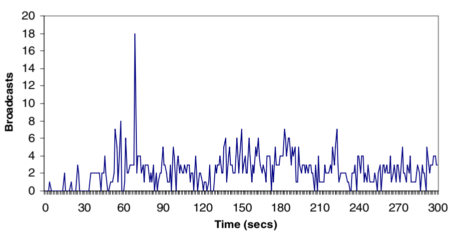


Fig. 7 c). 1st prove number of broadcasts

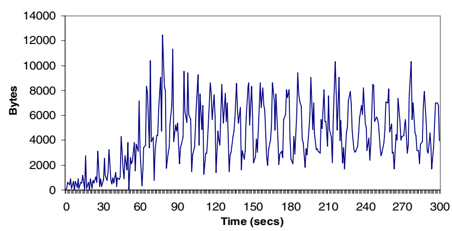


Fig. 8 a). 2nd prove bandwidth utilization

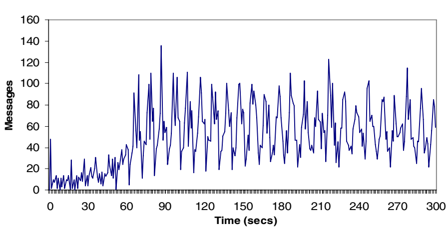


Fig. 8 b). 2nd prove number of messages

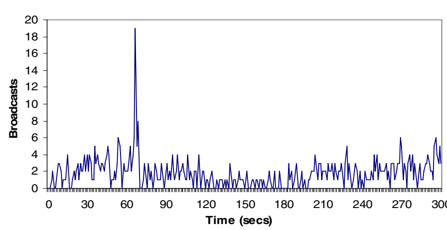


Fig. 8 c). 2nd prove number of broadcasts

When we increase the keepalive time, peaks values are lower and there are less bits per second and messages inside the network, but the time to check a node failure increases. We have observed that when we increase the number of groups in the network, but maintaining the number of nodes constant, the number of broadcast messages is almost the same. Although the number of nodes in the architecture is increased, there is not any proportion with the number of messages sent. If we cause to fail a DN with many connections with DNs from other groups, we can observe that the number of messages increased is not so significant to be seen in the graphs having a quick look. It is needed many DNs to have higher impact in the graphs.

7 Conclusions

We have presented a Grid architecture based on groups that is able to self-organize connections between nodes from different groups based on their available capacity. It is based on three types of roles for nodes of the architecture and their role is based on a promotion parameter. ACs organize DNs in zones to have a scalable architecture and help to establish DN connections routing DN information inside the group and choosing DNs with highest capacity. DNs have connections with DNs from other groups to share resources, data or computing time between groups. GCs have connections with GCs from other groups allowing groups interconnection and helping to organize DN connections. This design allows changing nodes' connections based on the available adjacencies and load from other ASs or DNs. Once the connections are established, to share resources, data or computing time between groups could be done without using ACs and GCs because they are used only for organization purposes. We have chosen SPF algorithm to reduce the latency to request new DNs when there are DN failures or leavings.

We have presented the analytical model and show the number of DNs in the network related with the number of regular nodes for several types of topologies. We have described the protocol operation and the recovery algorithm when any type of node leaves the architecture or fails down. The protocol does not consume so much bandwidth. We have shown that messages with more bandwidth are the backup messages and the one which sends the topological database, so, they are maintained by incremental updates. Real measurements demonstrate it is a feasible architecture because of the bandwidth consumption to manage the system is low and it can be used as an intergrid protocol or to replicate data from a group to other groups.

As future work, we will do some experimental results to adjust δ and λ parameters. On the other hand, we will test very short keepalive time and holdtime in order to reduce convergence times and to have a fast recovery algorithm for critical systems.

References

1. Foster, I., Kesselman, C., and Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 15(3):200 - 222, 2001.
2. García-Carballeira, F., Carretero, J., Calderón, A., García, J. D., Sanchez, L. A global and parallel file system for grids. Future Generation Computer Systems 23 (2007). Pp. 116–122.

3. J.G. Jensen, T. Shah, O. Synge, J. Gordon, G. Johnson, R. Tam, Enabling Grid Access to Mass Storage: Architecture and Design of the EDG Storage Element. Journal of Grid Computing 3 (1–2) (2005) 101–112.
4. Ruay-Shiung, C. and Po-Hung, C. Complete and fragmented replica selection and retrieval in Data Grids, Future Generation Computer Systems. In Press.
5. Ping, L., Kevin, L., Zhongzhi, S. and Qing, H., Distributed data mining in grid computing environments, Future Generation Computer Systems, Vol. 23, Issue 1. Jan 2007. Pp. 84-91
6. Koo, S. G. M., Kannan, K., Lee, C.S.G., A genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks, Proc. of the 13th IEEE International Conference on Computer Communications and Networks, Chicago, IL, October 2004, pp. 469–474.
7. Ferreira, R. A., Jagannathan, S., Grama, A. Locality in structured peer-to-peer networks, Journal of Parallel and Distributed Computing, Vol. 66, Issue 2. Feb. 2006. Pp. 257-273.
8. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J. Routing with guaranteed delivery in ad hoc wireless networks, Wireless Networking 7 (6) (2001). Pp. 609–616.
9. Xu, Z., Tang, C., Zhang, Z. Building topology-aware overlays using global soft-state. Proc. of the 23rd International Conference on Distributed Computing Systems, 2003. May 2003.
10. Wierzbicki, A., Strzelecki, R., Swierczewski, D. and Znojek, M. Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications, in: Second IEEE International Conference on Peer-to-Peer Computing (P2P2002), Linöping, Sweden, 2002.
11. Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y. Peer-to-Peer Based Multimedia Distribution Service, IEEE Transactions on Multimedia 6 (2) (2004).
12. Hongjun, L., Luo, L. P. and Zhifeng, Z. A structured hierarchical P2P model based on a rigorous binary tree code algorithm, Future Generation Computer Systems 23 (2). 2007. Pp. 201-208.
13. Lloret, J., Boronat, F., Palau, C., Esteve, M.: Two Levels SPF-Based System to Interconnect Partially Decentralized P2P File Sharing Networks, International Conference on Autonomic and Autonomous Systems International Conference on Networking and Services Joint ICAS'05 and ICNS'05, (2005).
14. Moy, J.: RFC 1245 - OSPF Protocol Analysis (1991). Available at http://www.faqs.org/rfcs/rfc1245.html
15. Lloret, J., Diaz, J. R., Jimenez, J. M., Boronat, F.: An Architecture to Connect Disjoint Multimedia Networks Based on node's Capacity, Lecture Notes in Computer Science, Vol. 4261. Springer-Verlag, Berlin Heidelberg New York (2006). Pp. 890-899.
16. Siganos, G., Faloutsos, M., Faloutsos, P. and Faloutsos, C. Power Laws and the AS-Level Internet Topology. IEEE/ACM Transactions on Networking, Vol. 11, Issue 4. August 2003.
17. Huberman, B.A. and Adamic, L.A. Growth dynamics of the World-Wide Web, Nature, vol. 40, (1999). pp. 450-457.
18. Ge, Z., Figueiredo, D. R., Jaiswal, S., Kurose, J., Towsley, D. Modeling Peer-Peer File Sharing Systems, Proceedings IEEE INFOCOM 2003, San Francisco, March-April 2003.
19. Hermann, G. Mathematical investigations in network properties. - Proceedings of the IEEE International Conference on Intelligent Engineering Systems. Pp 79-82. September 2005.
20. D. J. Watts and H. S. Strogatz, Nature 393, 440 (1998).
21. Cramer, C., Kutzner, K. and Fuhrmann T. Bootstrapping Locality-Aware P2P Networks. The IEEE International Conference on Networks, Vol. 1. (2004). Pp. 357-361.

A Group-Based Architecture for Wireless Sensor Networks

Jaime Lloret, Miguel Garcia and Jesus Tomas
Department of Communications
Polytechnic University of Valencia
Valencia, Spain
jlloret@dcom.upv.es, migarpi@teleco.upv.es, jtomas@dcom.upv.es

Abstract—**Many routing protocols for ad-hoc networks and sensor networks have been designed, but none of them are based on groups. We propose to divide the network into several groups of sensors. When a sensor send data to other groups, the data has to arrive just to one sensor from each group, then they propagate it to the rest of sensors in their groups. We have simulated our proposal for different types of sensor topologies to know which type of topology is the best depending on the number of sensors in the whole network or depending on the number of interior sensors. We have also simulated how much time is needed to propagate information between groups. The application areas for our proposal could be rural and agricultural environments to detect plagues and to propagate it to neighbouring areas, or for military purposes to propagate information between neighbouring squads.**

Keywords-Sensor Network; Group-Based Architecture; Group-based routing algorithm.

I. INTRODUCTION

There are many routing protocols that can be applied to sensor networks. They can be classified into two groups [1] [2]. One group is formed by protocols based on the network topology and the other group the ones that do not take it into account. First group can be broken down into three subgroups:

1- Plane routing. All nodes in the network have the same role and perform the same tasks. Because of the number of nodes in these networks, the use of a global identifier, for every node, is not feasible. It uses a data-centric routing where the base station sends requests to some regions and the nodes from that regions reply. Some of the algorithms in this group are SPIN, Direct diffusion, Rumour routing, MCFA, GBR, IDSQ, CADR, COUGAR, ADQUIRE, and so on.
2- Hierarchical routing. It is very scalable and has an efficient communication. It has been designed for energy saving purposes, because central nodes have unlimited energy, while leaf sensors have limited energy. When the sensor network topology is formed, data can be routed. Some algorithms such as LEACH, PEGASIS, TEEN, APTEEN, MECN, Virtual grid architecture routing and TTDD are hierarchical routing algorithms.
3- Position-based routing. All data is routed through the sensors depending on their position. Distances between sensors are known because of neighbouring sensors signals. There are other protocols that base node's situation on GPS and, using that information, route the data to the most adequate sensor. These algorithms consume more energy than others because of the need of GPS signal. Some of those algorithms sleep sensors when the network has not any activity. Some examples are GAF, GEAR, GOAFR and SPAN.

Second group does not have into account the structure of the network. It can be broken into five subgroups:

1- Multipath Routing Protocols. The information could reach the destination through different paths. Because sensors have to calculate several paths, they use a main route when they have enough energy; otherwise, they use an alternative path.
2- Query-Based Routing protocols. They are based on a central node that sends a query about an event to the specific area. When the query arrives to that area, it is routed to the destination sensor, and then it will reply. A sensor from an area could be sleeping, saving energy, while there is not any query to that area.
3- Negotiation-Based Routing Protocols. Before data transmission, the sensor has to negotiate the data it has to send, so redundant data could be deleted, and resources will be available while data exchange. SPIN protocols use this type of routing, but they take into account the network structure.
4- QoS protocols. The information is routed to the sensors taking into account quality parameters such as delay, energy, bandwidth and so on. SAR and SPEED protocols are based on quality of service algorithms.
5- Data coherent/incoherent processing based protocols. These algorithms use several routing techniques taking into account the data processing of a coherent or incoherent result.

None of the routing protocols aforementioned are group-based. We propose to divide the network of sensors into several groups and if a sensor has to send data to other groups, when this data arrives to one sensor from a group, it propagates it to the rest of sensors in its group.

The paper is structured as follows. Section 2 examines some works related with our proposal such as neighbour selection and architectures based on groups, and explains our motivation. There is a description of our architecture proposal in section 3. Analytical model for some types of topologies of sensors are shown in section 4. The propagation time to reach a sensor from other group is analyzed in section 5. Finally, section 6 gives our conclusions and future works.

Third International Conference on Networking and Services(ICNS'07)
0-7695-2858-9/07 $20.00 © 2007 IEEE

II. Previous works and motivation

Throughout the years, different types of strategies for neighbors' selection have been developed. On one hand, there are the ones used for transfer coordination to increase content availability. They can be applied for P2P networks [3], [4] and [5], for content delivery systems [6] or for distributing systems [7]. Many other systems locate nodes in the topology based on mathematical structures such as CAN, Chord, Pastry and Tapestry, but these systems do not take care of the underlying network, so a neighbor of a node could be very far (in terms of round-trip time –RTT-) or it could not have enough capacity available to perform its necessities. There are proposals where nodes' connections are based on the underlying network, such as Plethora [8] or on their geographic location such as the one described in [9]. Other systems locate new nodes in the topology taking into account that they are possibly close to a given node, and then, perform RTT measurements to identify the actual closest node such as the one presented in [10], and others use a proximity neighbor selection (PNS) using heuristics approximations such as the one presented in [11]. There are also some researches for wireless networks, where connections are established only if they are closed, because of their coverage area ([12] and [13]).

But none of the neighbor selection strategies shown consider to group nodes and structure connections between nodes from different groups. On the other hand, none of them take into account the capacity of the nodes to select the neighbor to have a connection with.

There are several works in the literature where nodes are grouped into groups and connections are established between nodes from different groups, but all of them have been developed to solve specific issues. Rhubarb [14] organizes nodes in a virtual network, allowing connections across firewalls/NAT, and efficient broadcasting. The nodes can be active, if they establish connections, or passive, if they don't. Rhubarb system has only one coordinator per group and coordinators could be grouped in groups in a hierarchy. The system uses a proxy coordinator, an active node outside the network, and all nodes inside the network make a permanent TCP connection with the proxy coordinator, which is renewed if it is broken by the firewall or NAT. If a node from outside the network wishes to communicate with a node that is inside, it sends a connection request to the proxy coordinator, who forwards the request to the node inside the network. Rhubarb has a three-level group's hierarchy. It may be sufficient to support a million nodes but when there are several millions of nodes in the network it could not be enough, so it suffers from scalability problems. On the other hand, all nodes need to know the IPs of the proxy coordinator nodes to establish connections with nodes from other virtual networks. A Peer-to-Peer Based Multimedia Distribution Service has been presented in [15]. That paper proposes a topology-aware overlay in which nearby hosts or peers self-organize into application groups. End hosts within the same group have similar network conditions and can easily collaborate with each other to achieve QoS awareness. When a node in this architecture wants to communicate with a node from other group, the information is routed through several groups until it arrives to the destination but this solution only can be applied to logical networks because of neighboring nodes could be so far. There are other architectures based on super-peer models such as Gnutella 2 and FastTrack networks. Each super-peer in these networks creates a group of leaf nodes. Superpeers perform query processing on behalf of their leaf nodes. A leaf node sends the query to its superpeer that floods it to its superpeer neighbors up to a limited number of hops. The main drawback of this architecture is that all information has to be routed through the superpeer logical network. Finally, there are some hierarchical architectures were nodes are structured hierarchically and some parts of the tree are grouped into groups such as the ones presented in [16] and in [17]. In some cases, some nodes have connections with nodes from other groups although they are in different layers of the tree, but in all cases, the information has to be routed through the hierarchy to achieve nodes from other groups, so all layers of the hierarchy could be overloaded in case of having many data to be transferred.

Let's suppose we need to divide the network into groups or areas because of the physical implementation of the sensor network or for scalability purposes. All architectures previously shown don't solve that problem efficiently, because in the case of centralized architectures, the server will have many wireless connections at the same time, so it will need many resources. On the other hand, there is a central point of failure and a bottleneck. In the case of fully distributed architectures, it is very difficult to control the system and it needs much time to process tasks, because of the time needed to reach far nodes, decreasing the performance of the whole system.

III. Architecture Description

Our proposal is based on the creation of groups of sensors with the same functionality in the network. There is a central sensor that limits the zone where the sensors from the same group will be placed, but its functionality will be the same that the rest of the sensors. A sensor knows in which group is because it is given manually or by GPS.

When there is an event in one sensor, this event is sent to all sensors in its group. All nodes in a group know all information of their group. Border sensors are those sensors of the border of the group, and they have connections with border sensors from other groups as it is shown in figure 1.

Border sensors are used to send information to other groups or to receive information from other groups and distribute it inside. When a sensor has to send some information to its group and to neighboring groups, the information is forwarded using Reverse Path Forwarding (RPF) Algorithm [18] (each group has one RPF database), but when the information has to be sent to other groups only, the information is routed directly to the border sensor closest to that group. When the sensor from the neighbor group receives that information, it routes it to all nodes in its group. Because the system is based on groups, the information is forwarded very fast to other groups (the information is routed through the shortest path to the border area sensor). Connections between border sensors from different groups are established as a function of their available processing capacity, their available number of connections, their available power or because a neighbor sensor failure. Figure 2 shows a logical view of the proposed architecture.

Third International Conference on Networking and Services(ICNS'07)
0-7695-2858-9/07 $20.00 © 2007 IEEE

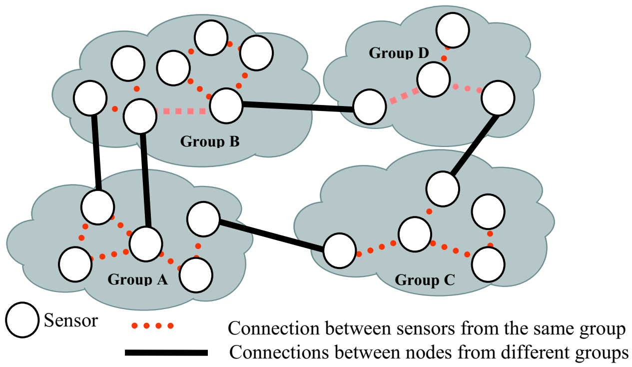


Figure 1. Topology example

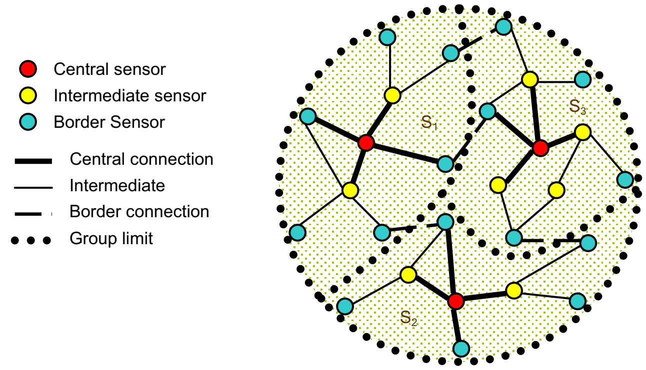


Figure 2. Logical view of the proposed architecture

IV. ANALITICAL MODEL

This section describes the architecture analytically taking into account that it is a system based on groups. Now, we are going to analyze the architecture for several types of network architectures inside the groups.

Let a network of sensors G = (V, λ, E) be, where V is the set of sensors, λ is the set of their capacities ($\lambda(i)$ is the capacity of the i-th sensor and $\lambda(i)\neq 0\ \forall$ i-th sensor) and E is the set of connections between sensors. Let k be a finite number of disjoint subsets of V, so $V=\cup V_k$ and there is not any sensor in two or more subsets ($\cap V_k=0$). Let's suppose n=|V| (the number of sensors in V) and k the number of subsets of V. We obtain equation 1.

$$n=\sum_{i=1}^{k}|V_k| \tag{1}$$

Every V_k has a central sensor, several intermediate sensors and several border sensors as it is shown in expression 2.

$$n = 1 + n_{intermediate} + n_{border} \tag{2}$$

Now we can describe the whole network as the sum of all these sensors from all groups as it is shown in equation 3.

$$n=\sum_{i=1}^{k}|(n_{central}+n_{\text{intermediate}}+n_{border})_k| = k+\sum_{i=1}^{k}(|n_{\text{intermediate}}|)_k+\sum_{i=1}^{k}(|n_{border}|)_k \tag{3}$$

Now we are going to model our proposal as a function of the number of intermediate and border sensors in a network for several types of networks.

A. Tree topology

Tree topologies have a sensor acting as a trunk and from this sensor leaves several branches. There are two types of tree topologies: N-nary trees (every sensor has the same number of leaf nodes, binary, ternary and so on) and backbone trees, where there is a trunk and there are sensors that branch from it. In both cases the information flows hierarchically. We are going to study the first case only, because it could be easily implemented by limiting the number of incoming connections in a sensor. The backbone tree is a special case of the partially centralised P2P Networks with superpeers and it will be discussed later.

In a tree topology, the number of sensors n is equal to $M^k - 1$, where M=2 in case of a binary tree, M=3 in a ternary tree and so on, and k is the number of levels of the tree). The number of links is n-1 and the diameter of the network is 2•k-2. We suppose balanced trees where all branches have the same number of levels, so the number of intermediate sensors is given by expression 4.

$$n_{\text{intermediate}}=\frac{n-1}{grade}-1 \tag{4}$$

Where $grade$ is the number of leaf sensors for each sensor. Using expressions 2 and 4, we obtain expression 5. It gives the number of border sensors related with the number of intermediate sensors.

$$n_{border} = (grade-1)\bullet n_{intermediate} + grade \tag{5}$$

Tree topologies have been implemented in several sensor networks such as the one shown in [19].

B. Grid topology

We are going to consider 2-dimensional Grid and 3-dimensional Grid with all its sides equals. To make easy the mathematical development, in a 2D Grid we will use a square matrix where $n=m$ for $n\geq 3$ and in a 3D Grid we will use a cube matrix where $n=m=l$ for $n\geq 3$. In both cases, the case of n=3 has one central sensor, but there is not any intermediate sensor.

The number of sensors in a 2D Grid, with all sides equals, sensor network is n^2 (n = 3, 4...). The number of neighbours of an intermediate sensor is 4, the border sensor has 3 neighbours and the vertex sensor has 2 neighbours. The number of connections in the topology is given by expression 6.

$$l=2\cdot\left(n-\sqrt{n}\right) \tag{6}$$

Expression 7 gives the diameter of a 2D Grid topology.

$$d=2\cdot\left(\sqrt{n}-1\right) \tag{7}$$

We have observed that the number of border sensors in a 2D Grid topology follows the expression 8.

$$n_{border}=4\cdot(\sqrt{n}-1) \tag{8}$$

Using expression 2, we obtain expression 9. It gives the number of intermediate sensors.

$$n_{\text{intermediate}}=n-4\left(\sqrt{n}-1\right)-1 \tag{9}$$

Third International Conference on Networking and Services(ICNS'07)
0-7695-2858-9/07 $20.00 © 2007 IEEE

Using expressions 8 and 9, we obtain expression 10 that relates the number of border sensors related with the number of intermediate sensors.

$$n_{border} = 4\cdot\left(1+\sqrt{n_{\text{int}\,ermediate}+1}\right) \tag{10}$$

2D Grid topologies have been implemented in several sensor works such as the one shown in [20].

The number of sensors in a 3D Grid, with all sides equals, sensor network is n^3 (n = 3, 4...). The number of neighbours of an intermediate or central sensor is 6, border sensors have 5 neighbours and the vertex sensors have 4 neighbours. Expression 11 gives the number of connections in the topology.

$$l = 3\cdot\left(n-\sqrt[3]{n^2}\right) \tag{11}$$

Expression 12 gives the diameter of a 3D Grid topology.

$$d = 3\cdot\left(\sqrt[3]{n}-1\right) \tag{12}$$

The number of border sensors in a 3D Grid topology can be measured by expression 13.

$$n_{border} = 6\cdot\sqrt[3]{n^2}-12\cdot\sqrt[3]{n}+8 \tag{13}$$

Using equation 2, we can obtain the number of intermediate sensors in a 3D Grid topology.

$$n_{\text{int}\,ermediate} = n-6\cdot\sqrt[3]{n^2}+12\cdot\sqrt[3]{n}-9 \tag{14}$$

Using the cube geometry, we can obtain the number of border sensors as a function of the number of intermediate sensors. This relation is given by equation 15.

$$n_{border} = 6\cdot\sqrt[3]{\left(n_{\text{int}\,ermediate}+1\right)^2}+12\cdot\sqrt[3]{\left(n_{\text{int}\,ermediate}+1\right)}+8 \tag{15}$$

3D Grid topology is used in networks that need many paths to reach the same destination.

C. Power Law

In [21], M. Faloutsos et al. show that the nodes of a distribution network can be modelled using mathematical laws. This paper states that power law fits real measurements with correlation coefficients of 96%. Power law states that the grade of a node (d_v) is proportional to its range (r_v) to the power of a constant called R as it is shown in expression 16.

$$d_v \propto r_v^{\,R} \tag{16}$$

Where R varies depending on it is applied. Applying Lemma 1, from paper [21], the grade of a node is given by expression 17.

$$d_v = \frac{1}{n^R}\cdot r_v^{\,R} \tag{17}$$

Where n is the number of sensors in the network, d_v and r_v are the grade and the range of the v sensor respectively.

From the power law appears the Zipf's law. It states that some nodes have many links while many nodes have one or two links. Zipf's law has been proposed by B. A. Huberman et al. to model Internet in [22], and by Z. Ge et al. to model Gnutella and Napster Networks in [23].

Zipf 's function states that the range of r nodes follows the proportionality shown in expression 18.

$$f(r) = C\cdot r^{-\alpha} \tag{18}$$

Where α varies depending on the type of distribution of the nodes. It is also known as the Zipf coefficient. C is a constant that varies depending on the type of network.

Taking into account expressions 17 and 18, we can assume that $R=-\alpha$. Applying Zipf's law to our sensor architecture, we obtain expression 19.

$$n_{border} = \frac{n^{\alpha}}{\left(n_{\text{int}\,ermediate}+1\right)^{\alpha}} \tag{19}$$

Taking expression 2 into account, we can obtain expression 20. It relates the number of border sensors with the total number of sensors in the topology.

$$n = \frac{\left(n_{border}\right)^{\frac{\alpha+1}{\alpha}}}{\left(n_{border}\right)^{1/\alpha}+1} \tag{20}$$

On the other hand, replacing expression 2 in expression 19 we obtain the number of border sensors as a function of the number of intermediate sensors as it is shown in expression 21.

$$n_{border} = \frac{\left(n_{border}+n_{\text{int}\,ermediate}+1\right)^{\alpha}}{\left(n_{\text{int}\,ermediate}+1\right)^{\alpha}} \tag{21}$$

As Internet topology has varied along the years, because of the growth of the number of computers connected to it, α value has varied from 0.74 to 3 in last measures, as it can be seen in [21] and [24].

D. Logarithmic law

Logarithmic law was introduced by György Hermann in [25]. This law proposes that the border nodes, or the nodes with higher roles in the network, are the responsible of the stability of the network. It also proposes that the border nodes are the responsible of the security of the network because they are the ones that communicate with exterior nodes. This proposal follows the model developed by D. J. Watts et al. in [26], where connections are established based on efficiency, stability and security features.

This law states that the distance between two border sensors is given by expression 22.

$$l_n \approx l_{\max} \approx \ln(n_{\text{int}\,ermediate}+1) \tag{22}$$

Where l_{max} is the diameter of the network. It is equal to the logarithm of the nodes that don't are in the border of the network (the central sensor plus the intermediate sensors).

The relationship between the number of border sensors and the intermediate sensors is given by expression 23.

$$c\cdot\left(n_{\text{int}\,ermediate}+1\right)\le n_{border}\le c\cdot\left(n_{\text{int}\,ermediate}+1\right)\cdot\ln(n_{\text{int}\,ermediate}+1) \tag{23}$$

C is a constant that depends on the model of the network.

So, the number of sensors in the network is set between limits shown in equation 24.

$$\begin{cases} n_{border} = \dfrac{n}{2} & \text{When } n_{min} \\ n = (n_{intermediate}+1)\cdot(1+\ln(n_{intermediate}+1)) & \text{When } n_{max} \end{cases} \quad (24)$$

E. Partially centralized P2P networks

In [27], J. Lloret et al. proposed an architecture for partially centralized P2P networks. They measured the number of brokers or superpeers (depending on the type of network), that was inside the architecture on behalf of all brokers or superpeers in the whole network. Those values could be applied to the proposal presented in this paper if we suppose that the intermediate sensors plus the central one are the distribution nodes and the border sensors are the nodes considered in the access layer. The relationship between intermediate sensors and border sensors are different according on the type of P2P network as it is shown in expression 25.

$$n_{border} = \begin{cases} n_{intermediate} + 1 & \text{in a broker model} \\ 96\cdot(n_{intermediate} - 1) & \text{in a superpeer model} \end{cases} \quad (25)$$

Using expression 2 we obtain expression 26.

$$n_{border} = \begin{cases} \dfrac{n}{2} & \text{in a broker model} \\ \dfrac{96\cdot(n-2)}{97} & \text{in a superpeer model} \end{cases} \quad (26)$$

F. Architectures comparation.

This section compares the number of border sensors versus the number of intermediate sensors and the number of border sensors versus the number of sensors in the group for all architectures shown. In both cases, partially centralized P2P networks with brokers model is the same case than the minimum values of the logarithmic model.

Figure 3 graphs the number of border sensors in the group as a function of the number of intermediate sensors for all models previously analyzed. For Zipf's law we have used numerical methods to obtain its graph. In figure 6, we can observe that if a group with few border sensors is needed, if there are less than 24 intermediate sensors, the best election is the minimum value of the logarithmic law, but if we have more than 24 intermediate sensors the best one is 2D Grid. What is desirable is to have many border sensors in order to have many connections with sensors from other groups, so there will be higher probability to contact with more neighbouring groups. We have checked that for less than 770 intermediate sensors the best topology is the partially centralized P2P networks with superpeer model, but if the number of intermediate sensors is equal or higher that 770, the best topology is Zipf's law with R=-2.45.

Figure 4 shows the number of border sensors in the group as a function of the number of sensors in the group. We have used numerical methods to know the number of border sensors as a function of the number of sensors in the group for the logarithmic model and for Zipf's law. In figure 4, we can observe that, when many border sensors are needed versus the number of sensors in the group, for less than 40 sensors the best election is 3D Grid, but for 40 sensors or more, the best election is the partially centralized network with superpeers model. When we need few border sensors versus the number of sensors in the group, for less than 110 sensors the best topology is the ternary tree, but for more than 110 sensors the best topology is 2D Grid.

V. Propagation Time

Every time a sensor has to send information to a specific group, first it has to send the information to the border sensor closest to that group, and then, the information has to be sent through the groups till the information arrives to the destination group. Expression 27 formulates it mathematically.

$$T = t_{to\_border} + \sum_{i=1}^{n} t_{max\_intragroup\_i} + \sum_{i=1}^{n+1} t_{border\_i-border\_i+1} \quad (27)$$

Where t_{to_border} is the time needed to reach the border sensor closest to that group, n is the number of intermediate groups through that path, $t_{max\_intragroup\_i}$ is the time needed to cross the i-th group and $t_{border\_i-border\_i+1}$ is the time needed to transmit the information from one border sensor to another border sensor from other group.

Let's suppose that t_p is the mean value of the propagation time for all transmissions between 2 sensors in the architecture. So, we can assume that $t_{border\_i-border\_i+1} = t_p$ and, given d_1 hops to reach from a source sensor to the border sensor closest to the destination group, we can assume $t_{to\_border} = d_1 \cdot t_p$. We can define the time needed to cross the i^{th}-group as $t_{max\_intragroup\_i} = d_i \cdot t_p$, where d_i is the number of hops to cross the i^{th}-group. Expression 28 gives the time needed to reach a group as a function of the mean value of the propagation time.

$$T = \left(d_1 + \sum_{i=1}^{n} d_i + n + 1\right)\cdot t_p \quad (28)$$

Let's consider four groups along a path to a group destination. Figure 5 shows two simulations. The first one (source group) shows the time needed when the mean value of the number of hops to cross the groups involved in the path is 10 and the number of hops from the source sensor to the border sensor closest to the destination group vary from 1 to 32. The second one (mean value of groups) shows the time needed when the number of hops from the source sensor to the border sensor closest to the destination group is 10 and the mean value of the number of hops to cross the groups involved in the path vary from 1 to 32. In figure 5, we can observe that the delay is higher when the mean value of the number of hops in the groups increases, but it is less significant when the number of hops from the source sensor to the border sensor closest to the destination group increases.

VI. Conclusions

To the extent of our knowledge, there is not any previous interconnection system to structure connections between groups of nodes like the one presented in this paper. This paper demonstrates that it is a feasible option and it is independent of the structure of the sensors of the group, but some group architectures perform better than others. It could be applied to specific environments such as rural environments or for military purposes. We are now designing its fault-tolerance.

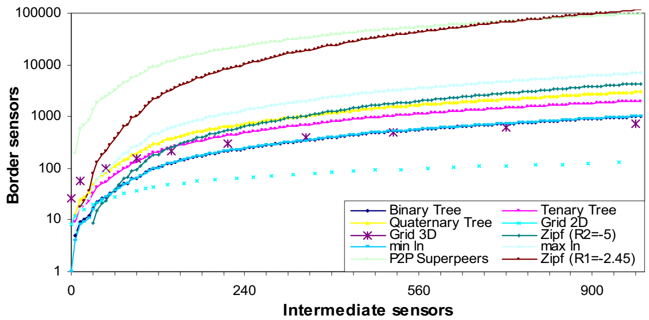


Figure 3. Border sensors as a function of the intermediate sensors

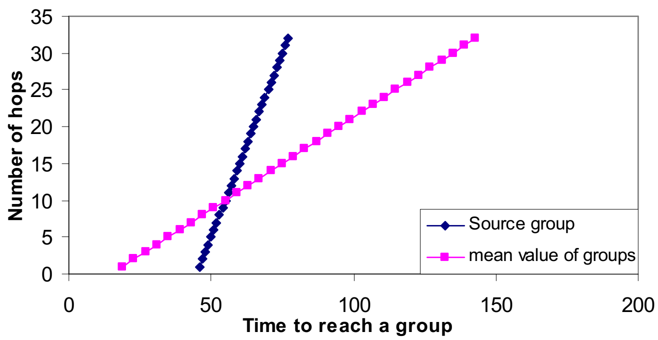


Figure 5. Time to reach a group as a function of the number of hops

ACKNOWLEDGMENT

Authors thank Mr. Antonio Canos, from the Polytechnic University of Valencia, because of his support obtaining, using numerical methods, the inverse of some functions used here.

REFERENCES

[1] L.M. Feeney, A taxonomy for routing protocols in mobile ad hoc networks, October 1999, SICS Technical Report T99:07.

[2] J. N. Al-Karaki, A. E. Kamal, Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications Volume 11, Issue 6, pp. 6-28, Dec. 2004.

[3] L. Zou, E. Zegura, M.H. Ammar, The effect of peer selection and buffering strategies on the performance of peer-to-peer file sharing systems, in: Proceedings of Tenth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS, Fort Worth, TX, 2002.

[4] Simon G.M. Koo, Karthik Kannan and C.S. George Lee, On neighbor-selection strategy in hybrid peer-to-peer networks, Future Generation Computer Systems 22 (7) (2006). Pp. 732-741.

[5] D.S. Bernstein, Z. Feng, B.N. Levine, S. Zilberstein, Adaptive peer selection. Proceedings of the 2nd International Workshop on Peer-to-Peer Systems, IPTPS03, Berkeley, CA, February 2003.

[6] J. Byers, J. Considine, M. Mitzenmacher, S. Rost, Informed content delivery across adaptive overlay networks, in: Proceedings of ACM SIGCOMM 2002, Pittsburgh, PA, August 2002.

[7] S.G.M. Koo, C.S.G. Lee, K. Kannan, A resource-trading mechanism for efficient distribution of large-volume contents on peer-to-peer networks, Proc. of the 14th IEEE International Conference on Computer Communications and Networks, San Diego, CA, October 17–19, 2005, pp. 428–433.

[8] Ferreira, R. A., Jagannathan, S., Grama, A. Locality in structured peer-to-peer networks, Journal of Parallel and Distributed Computing, Vol. 66, Issue 2. Feb. 2006. Pp. 257-273.

[9] Bose, P., Morin, P., Stojmenovic, I., Urrutia, J. Routing with guaranteed delivery in ad hoc wireless networks, Wireless Networking 7 (6) (2001). Pp. 609–616.

[10] Xu, Z., Tang, C., Zhang, Z. Building topology-aware overlays using global soft-state. Proc. of the 23rd International Conference on Distributed Computing Systems, 2003. May 2003.

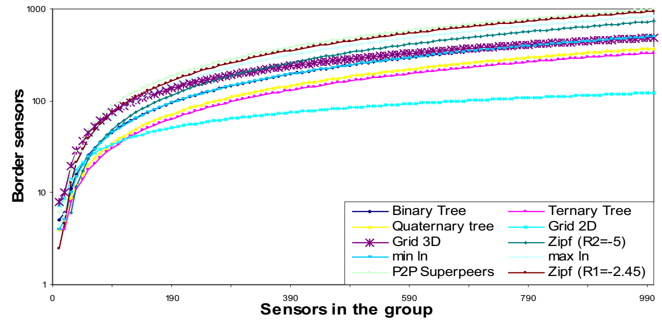


Figure 4. Border sensors as a function of the number of sensors in the group

[11] Castro, M., Druschel, P., Hu, Y. C. and Rowstron, A. Proximity neighbor selection in tree-based structured peer-to-peer overlays, Technical Report MSR-TR-2003-52, Microsoft Research, Microsoft Corporation. 2003.

[12] Karp, B., Kung, H.T. GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom '00), ACM Press. New York, NY, USA, 2000, Pp. 243–254.

[13] Kim, Y. J., Govindan, R., Karp, B., Shenker, S. Geographic routing made practical, in: Proceedings of the Second USENIX/ACM Symposium on Networked System Design and Implementation (NSDI'05), May 2005, pp. 217–230.

[14] Wierzbicki, A., Strzelecki, R., Swierczewski, D. and Znojek, M. Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications, in: Second IEEE International Conference on Peer-to-Peer Computing (P2P2002), Linöping, Sweden, 2002.

[15] Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y. Peer-to-Peer Based Multimedia Distribution Service, IEEE Transactions on Multimedia 6 (2) (2004).

[16] Hongjun, L., Luo, L. P. and Zhifeng, Z. A structured hierarchical P2P model based on a rigorous binary tree code algorithm, Future Generation Computer Systems 23 (2). 2007. Pp. 201-208.

[17] Thallner, B., Moser, H. Topology control for fault-tolerant communication in highly dynamic wireless networks. Proceedings of the Third International Workshop on Intelligent Solutions in Embedded Systems. May 2005.

[18] Yogen K. Dalal y Robert M. Metcalfe, "Reverse path forwarding of broadcast packets", Communications of the ACM. Volume 21, Issue 12 Pp: 1040 – 1048. Diciembre 1978. I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[19] Jan Beutel, Matthias Dyer, Martin Hinz, Lennart Meier and Matthias Ringwald. Next-generation prototyping of sensor networks, Proceedings of the 2nd international conference on Embedded networked sensor systems. Pp: 291 – 292. 2004.

[20] Radu Stoleru and John A. Stankovic, Probability Grid: A Location Estimation Scheme for Wireless Sensor Networks, Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. Pp. 430-438. 4-7 Oct. 2004.

[21] Siganos, G., Faloutsos, M., Faloutsos, P. and Faloutsos, C. Power Laws and the AS-Level Internet Topology. IEEE/ACM Transactions on Networking, Vol. 11, Issue 4. August 2003.

[22] Huberman, B. A. and Adamic, L. A. Growth dynamics of the World-Wide Web, Nature, vol. 40, (1999). Pp. 450-457.

[23] Ge, Z., Figueiredo, D. R., Jaiswal, S., Kurose, J., Towsley, D. Modeling Peer-Peer File Sharing Systems, Proceedings IEEE INFOCOM 2003, San Francisco, March-April 2003.

[24] Albert, R., Jeong, H. & Barabási, A. L. Diameter of the World Wide Web. (1999) Nature 401, 130–131.

[25] Gyorgy Hermann. Mathematical investigations in network properties. Proceedings IEEE Intelligent Engineering Systems, 2005. INES '05. Pp 79-82 September 16-19, 2005

[26] D. J. Watts and H. S. Strogatz. Collective dynamics of 'small-world' networks. Nature 393, 440 (1998).

[27] Lloret, J., Boronat, F., Palau, C., Esteve, M.: Two Levels SPF-Based System to Interconnect Partially Decentralized P2P File Sharing Networks, Joint International Conf. on Autonomic and Autonomous Systems International Conf. on Networking and Services (2005).

Sincronización de grupo multimedia basada en protocolos estándar

Fernando Boronat Seguí, Juan Carlos Guerri Cebollada, Jaime Lloret Mauri, Miguel García Pineda
Universidad Politécnica de Valencia - Escuela Politécnica Superior de Gandia
Ctra. Nazaret-Oliva S/N, 46730 Grao de Gandia (VALENCIA)
Teléfono: +34 962 849 341, Fax: +34 962 849 309
E-mail: {fboronat, jcguerri, jlloret}@dcom.upv.es

Abstract. *Most of actual multimedia tools use RTP/RTCP for inter-stream synchronization, but not for group synchronization. A new proposal of modification of RTCP packets to provide a sender-based method for synchronization of a group of receivers is described and evaluated both objectively and subjectively. The solution takes advantage of the feedback RR RTCP messages and the malleability of RTP/RTCP to provide the information required by the synchronization approach, defining a few new APP RTCP packets useful for synchronization purpose. This modification hardly increases the workload of the network and helps to avoid the asynchronies, between receivers (distributed) and between streams (locally), exceeding the limits, in accordance with the related literature.*

1 Introducción

Actualmente, existen muchas aplicaciones multimedia distribuidas basadas en la cooperación (teleenseñanza, televigilancia, juegos en red, distribución de video con su audio en diferentes idiomas, etc.), las cuales incluyen la transmisión de diferentes flujos (audio, video, texto, datos,…), normalmente de forma multicast, desde una o varias fuentes a uno o varios receptores. Todas ellas incluyen normalmente sincronización intra-flujo (añaden algún mecanismo que garantice las relaciones temporales entre unidades de datos –LDUs o *Logical Data Units*- de un mismo flujo, como, por ejemplo, entre las tramas de una misma secuencia de video) e inter-flujo (garantizando las relaciones temporales entre las LDUs de los diferentes flujos multimedia, como, por ejemplo, la reproducción del audio de un discurso y los movimientos asociados de los labios del locutor del discurso, conocida como sincronización labial o *Lip-Sync*).

Sin embargo, en determinadas aplicaciones se necesita otro tipo de sincronización, denominado *Sincronización de Grupo*, que consiste en garantizar la reproducción sincronizada de todos los flujos tanto localmente (inter-flujo) en cada receptor como, a la vez y globalmente, en todos los receptores (en grupo). Se ocupa de garantizar la reproducción de todos los flujos de forma sincronizada en todos los receptores al mismo tiempo. Se han encontrado muy pocas soluciones incluyendo este tipo de sincronización, entre las que se pueden destacar [1], [2], [3] y [4], todas las cuales se basan en el receptor (*receiver-driven*) y, excepto la presentada en [3] que utiliza RTP/RTCP ([5]), ninguna utiliza protocolos estándar en sus propuestas sino que definen nuevos protocolos con mensajes de control de la sincronización específicos, que se intercambian entre las fuentes y los receptores para obtener la sincronización final deseada. Destacamos la solución presentada por Akyldiz y Yen en [2] y el algoritmo *VTR* (*Virtual Time Rendering*, [4]). Ambas soluciones también se basan en el receptor (*receiver-driven*), utilizan un receptor como referencia para la sincronización (esquema maestro/esclavo) e incluyen intercambio de información entre receptores para sincronizarse con el de referencia, lo cual implica una carga de red considerable. El algoritmo propuesto en [2] también propone un mecanismo para sincronizar el instante inicial de la reproducción en todos los receptores.

Por otro lado, también se han encontrado dos RFCs, la 4585 ([6]) y la 4586 ([7]) que definen nuevas extensiones para el perfil *Audio-visual Profile (AVP) for RTCP-based feedback (RTP/AVPF)*, que permite a los receptores proporcionar realimentación de forma más inmediata a las fuentes y así permitir una adaptación de la transmisión a corto plazo y la posibilidad de implementar mecanismos de recuperación. La RFC 4585 ([6]) también define un pequeño grupo de mensajes de realimentación RTCP de propósito general. Tal como se explica más adelante, en nuestra solución se han definido nuevas extensiones para determinados paquetes RTCP y, además, nuevos mensajes RTCP para realimentación útiles para el propósito de la sincronización de grupo deseada.

Se presenta un método novedoso para obtener la sincronización de grupo, basado en RTP/RTCP ([5]) y en NTP ([8]), minimizando el tráfico de control con respecto a las soluciones anteriores e incluyendo las técnicas más comunes de sincronización utilizadas por los algoritmos y soluciones más populares. En [9] se detallan dichas técnicas y se compara nuestra propuesta con dichas soluciones.

A continuación, en la sección 2 se expone la solución propuesta. En la sección 3 se muestran los resultados de los dos tipos de evaluación realizadas, finalizando el artículo con las conclusiones del mismo y las referencias bibliográficas.

2 Propuesta de Sincronización

La solución presentada será de aplicación en escenarios con sistemas distribuidos con una o varias fuentes de flujos multimedia transmitiendo, de forma multicast, y uno o varios receptores de dichos flujos, utilizando redes de comunicaciones determinísticas con unos requerimientos mínimos de calidad de servicio (al menos, deberá ser conocido o acotado el retardo extremo a extremo de la red). La estructura de la propuesta, en cuanto a funcionalidad, está basada en el protocolo *Feedback* ([10]), pero añadiendo la utilización de un tiempo global proporcionado por el protocolo NTP, tal y como se propone en el protocolo *Feedback Global* ([11]). En [10] se trabaja con relojes locales. Las soluciones propuestas en [10] y [11] sólo incluyen técnicas de sincronización intra e inter-flujo, son adaptativas, válidas para multicast, utilizan esquemas maestro/esclavo y técnicas de realimentación para intercambiar información entre fuentes y receptores.

Para resolver el problema de la sincronización en los receptores, dividimos el proceso en dos fases (Fig. 1):

1. Conseguir que todos los receptores inicien la reproducción de uno de los flujos, considerado como *flujo maestro*, en el mismo instante (*Instante Inicial de Reproducción*) y que, a partir de dicho instante, continúen la reproducción de dicho flujo de forma sincronizada (llamaremos a este proceso *sincronización distribuida de grupo entre receptores*).

2. Conseguir que localmente, en cada receptor, se reproduzcan de forma sincronizada todos los flujos que deba reproducir dicho receptor (*sincronización local inter-flujo*).

Para ello, nuestra propuesta se basa en dos *esquemas maestro/esclavo*. Por un lado, existirá un *receptor maestro* que servirá de referencia para la sincronización de grupo, entre receptores, y, por otro lado, existirá un *flujo maestro* que servirá de referencia para la sincronización inter-flujo interna en cada receptor.

En la Fig. 1 se puede apreciar la existencia de una transmisión, que puede ser *multicast* o *unicast*, de flujos multimedia mediante RTP desde una o varias fuentes transmisoras a uno o varios receptores. Uno de los flujos multimedia es tomado como flujo *maestro* (líneas y flechas de mayor grosor) y, además, de entre todos los receptores se selecciona uno de ellos como *receptor maestro* (gris en la figura), cuyo estado de reproducción del flujo *maestro* será tomado como referencia para determinar el estado de reproducción de cada uno de los demás receptores (*esclavos*). Este *receptor maestro* podrá ser elegido de varias maneras, según determinados criterios (tal y como se describe en [12]). Se utilizará RTCP para enviar mensajes de control durante la sesión.

La fuente transmisora del flujo maestro se convertirá en la *Fuente Sincronizadora* y será la que controlará que la reproducción de los receptores se haga de la forma más sincronizada posible, debiendo procesar y analizar la información de realimentación que estos le enviarán de forma, más o menos, periódica.

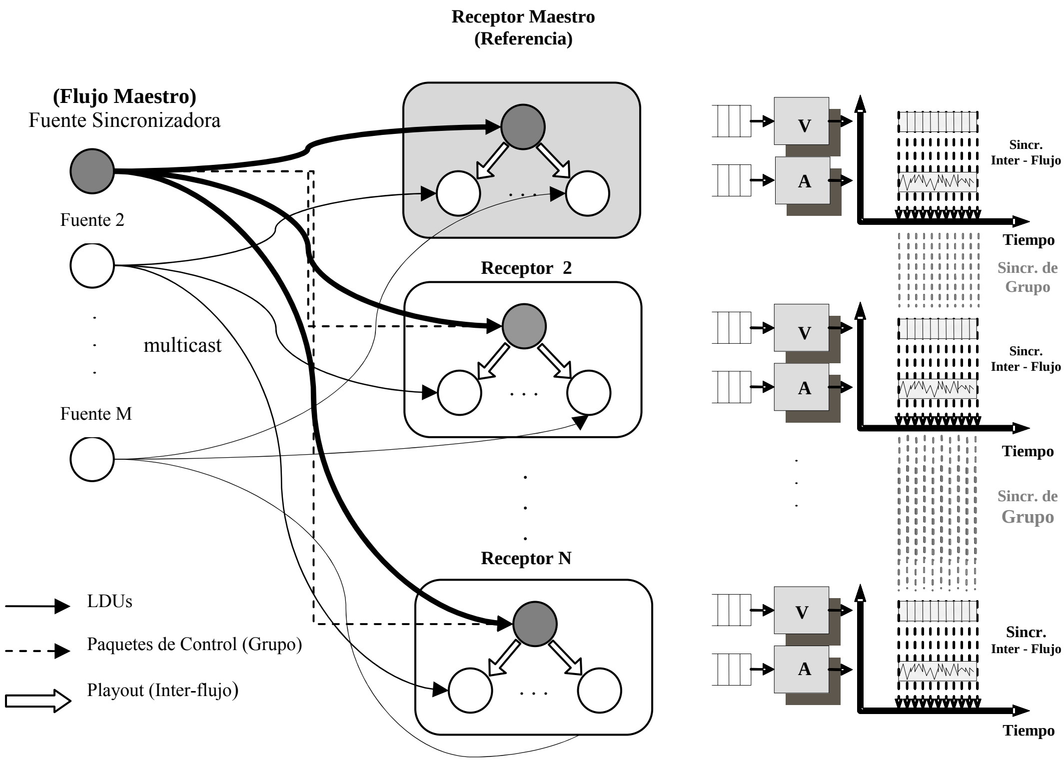


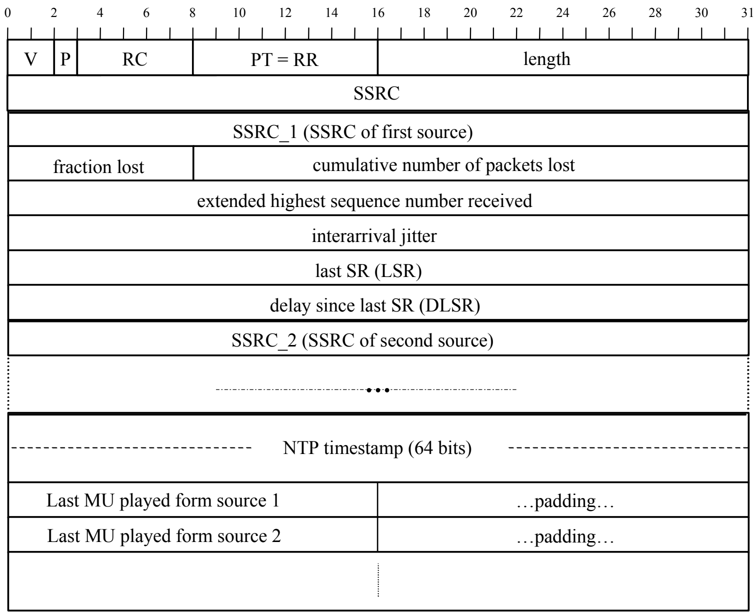
Figura 1. Sincronización Inter-flujo y de grupo

Los paquetes de realimentación en RTCP son los denominados *RTCP Receiver Reports* (paquetes RTCP RR, [5]) pero la información que contienen no es suficiente para nuestro propósito final de la sincronización. Es por ello que proponemos la modificación de dichos paquetes para incluir la información necesaria para dicho propósito. Además, se definirán nuevos paquetes RTCP APP (*Application-defined RTCP packet*, [5]) que utilizará la fuente para indicar cuándo se debe iniciar la reproducción y también las posteriores correcciones a los receptores en sus procesos de reproducción, cuando detecte que están entrando en situaciones de asincronía (paquetes que denominaremos '*paquetes de acción*').

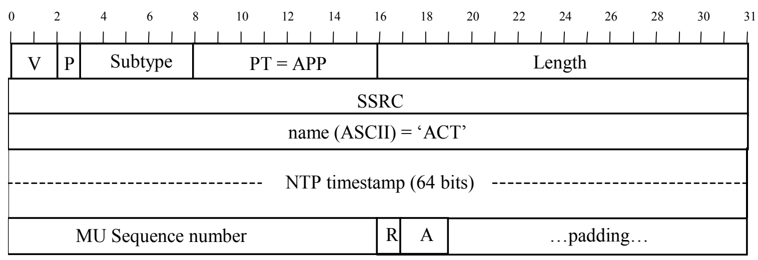
Bajo este punto de vista podríamos decir que nuestra solución, a diferencia de las comentadas anteriormente, está *basada en la fuente* (*source-driven*), ya que será la Fuente Sincronizadora la que, indirectamente, controlará los procesos de reproducción de los flujos en los receptores, a través del mecanismo de sincronización propuesto. Para ello tomará la información que le llegue de los paquetes RTCP RR modificados, la procesará y les enviará a los receptores paquetes de acción pertinentes.

La Fuente Sincronizadora necesita información de realimentación conteniendo el estado del proceso de reproducción del flujo maestro en cada receptor. Aprovechando las características de los protocolos RTP/RTCP, que pueden ser modificados para proporcionar la información requerida para una determinada aplicación, hemos definido nuevas extensiones de sus paquetes para contener la información necesaria.

Proponemos modificar el paquete *RTCP RR* ([5]), y llamarlo paquete *RTCP RR EXT* (de '*extendido*'), para incluir una extensión específica (*a profile-specific extension part*) a su formato, con la siguiente información: el número de la última LDU reproducida por el receptor y la marca de tiempo, en unidades NTP, del instante en que dicho receptor la reprodujo (Fig. 2a). Con esa información y una estimación de los límites del retardo de la red, la Fuente Sincronizadora puede conocer el estado de los procesos reproductores del flujo maestro en cada uno de los receptores (tal y como se explica en [12]). Una vez obtenida dicha información de cada receptor, tomará a uno de los receptores como referencia (considerado como *receptor maestro*, Fig. 1), calculará las asincronías entre el proceso de reproducción del flujo maestro del receptor maestro y los procesos de dicho flujo en los demás receptores y, a continuación, enviará (multicast) paquetes de acción para hacer que los receptores corrijan el estado de su proceso reproductor en consecuencia (los receptores retrasados respecto al receptor maestro, '*saltarán*' LDUs en su reproducción, mientras que los procesos reproductores adelantados repetirán la reproducción de la LDU que estén reproduciendo en ese instante, con el consiguiente efecto de 'pausa').



a) Paquete *RTCP RR EXT*



b) Paquete *RTCP APP ACT*

Figura 2. Formato de los paquetes propuestos

Para definir los *paquetes de acción* proponemos el uso de nuevos paquetes de control RTCP APP ([5]), que hemos denominado *paquetes RTCP APP ACT* (de '*acción'*), con una extensión dependiente de nuestra aplicación, incluyendo un número de secuencia de LDU y la marca de tiempo, en unidades NTP, del instante en que la LDU con dicho número de secuencia deberá ser reproducida por todos los receptores (Fig. 2b). Este paquete también servirá para indicar el instante de inicio común de la reproducción a todos los receptores de la primera LDU del flujo maestro.

El funcionamiento general del algoritmo propuesto es el mostrado en la Fig. 3, donde se representa la fuente sincronizadora (transmisora del flujo *maestro*) y los receptores *i* y *j* de la sesión.

Durante la sesión, la fuente sincronizadora irá recibiendo, de uno en uno, los paquetes *RTCP RR EXT* pertenecientes a todos los receptores que estén reproduciendo el flujo maestro transmitido por ella. De dichos paquetes extraerá la información relacionada con el identificador del receptor (identificador *SSRC*, definido en [5]), la última LDU reproducida por el mismo y el instante NTP en que dicho receptor reprodujo dicha LDU. Esta información se irá guardando en una tabla creada por la propia fuente sincronizadora con un número de registros igual al número de receptores participantes en la sesión (*n*), con la estructura mostrada en la tabla 1. En los casos en que la fuente reciba un segundo paquete *RTCP RR EXT* procedente de un mismo receptor antes de completar toda la tabla, actualizará la información, con el fin de mantener la tabla con los valores más recientes.

La columna '*bit de reproducción'* (R_i) indica si el receptor está o no activo y se utiliza para saber si el receptor incluido en la sesión está o no reproduciendo el flujo maestro y, por tanto, su información (LDU_i y NTP_i) deberá ser tomada en cuenta (bit a '1') o no (bit a '0') para realizar el cálculo del punto de reproducción de referencia. Este bit será necesario para poder considerar los abandonos de los receptores durante la sesión y evitar que los datos referentes a receptores no presentes en la sesión afecten al resto en un momento dado.

Una vez completada la tabla con los nuevos datos procedentes de todos los receptores activos, se estará en disposición de elegir a uno de los receptores como referencia siguiendo algún criterio específico (por ejemplo, el receptor más lento en su reproducción, o el más rápido, etc.).

Lo ideal, a la hora de calcular la referencia o receptor *maestro* con el cual se sincronizarán todos los demás receptores, sería que todos los receptores enviaran un paquete de control con dicha información a la vez, es decir, con la misma referencia temporal o instante NTP. Esto, lógicamente, en sesiones con un elevado número de usuarios podría suponer un envío masivo de paquetes de todos los receptores a la fuente en ciertos instantes, lo cual podría colapsarlo, afectando a la escalabilidad de la solución propuesta. Este ha sido uno de los motivos por los que se ha elegido el paquete RTCP RR para enviar la información necesaria descrita anteriormente. Tal y como describe la RFC 1889 ([5], en el Anexo 1, apartado 6.2, *Intervalo de transmisión RTCP*), cada receptor enviará su paquete de informe *RTCP RR EXT* de forma aleatoria. Por lo tanto, el momento NTP con el que los receptores enviarán sus paquetes *RTCP RR EXT* no será el mismo. Debido a esta aleatoriedad en el envío, la fuente se verá obligada a buscar una relación entre la última LDU consumida y el tiempo global y 'real' NTP. Esto es posible gracias a las marcas de tiempo NTP y RTP que contienen los paquetes RTCP.

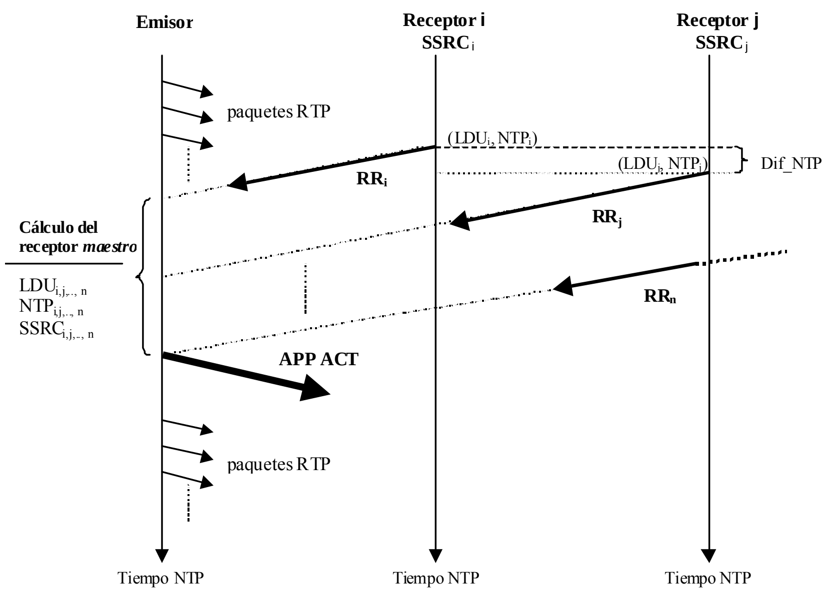


Figura 3. Funcionamiento General

Tabla 1. Información manejada por la fuente

SSRC	Última LDU	NTP timestamp	Bit de Reproducción R_i
$SSRC_1$	LDU_1	NTP_1	bit_1
$SSRC_2$	LDU_2	NTP_2	bit_2
$SSRC_n$	LDU_n	NTP_n	bit_n

Una vez conseguida la sincronización de grupo, es decir, cuando ya todos los receptores estén reproduciendo el flujo maestro de forma sincronizada, también será necesario un mecanismo adicional para conseguir que, localmente en cada receptor, los flujos que se reproduzcan en el mismo también lo hagan de forma sincronizada entre ellos (sincronización inter-flujo local). Para ello se hará uso de un bus interno de comunicación entre los procesos de reproducción del receptor, denominado *mbus* (cuya especificación está en [13]).

Mediante mensajes a través de *mbus* el proceso de reproducción del flujo maestro envía su estado de reproducción a todos los demás procesos reproductores de los flujos esclavos del receptor (Fig. 4) para que estos se adapten a dicho estado, mediante 'saltos' (o, lo que es lo mismo, descarte de las LDUs del buffer cuyo instante de reproducción ya haya pasado) y/o 'pausas' (lo que equivale a repetir la reproducción de la última LDU hasta que se deba reproducir la siguiente almacenada en el buffer de reproducción) en su reproducción.

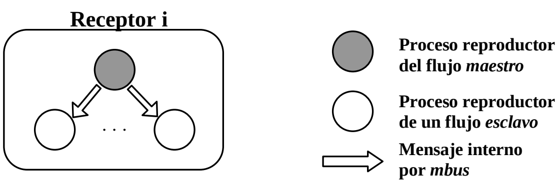


Figura 4. Sincronización local inter-flujo a través del bus interno *mbus*

En la Fig. 5 aparece el diagrama de flujos del intercambio de información entre los procesos del proceso reproductor del flujo maestro y el proceso reproductor de uno de los flujos esclavos. El proceso del flujo maestro le comunica al del flujo esclavo el valor de su *playout delay* en cada momento. Se trata del retardo de reproducción de la LDU que está reproduciendo en dicho instante, esto es, el retardo transcurrido desde que se transmitió dicha LDU desde la Fuente Sincronizadora hasta que es reproducida. Para evitar continuas adaptaciones, el proceso del flujo esclavo lo compara con el suyo propio y sólo hace correcciones si la diferencia entre los dos valores es superior a un determinado umbral que se configurará según las aplicaciones.

Ya que cada proceso reproductor de un flujo esclavo no tiene la misma referencia de reloj que el del flujo maestro, se hace uso de las marcas de tiempo NTP y del '*mapeado'* entre marcas RTP y marcas NTP para poder obtener una referencia común y así poder realizar la comparación de los valores del *playout delay*.

3 Evaluación

La propuesta ha sido implementada en una aplicación con dos flujos, uno de audio y otro de video, formada por herramientas Mbone, basadas en RTP, modificadas, como son *rat* ([14]), para transmisión multicast del flujo de audio, y *vic* ([15]), para la transmisión multicast del flujo de vídeo. Dicha aplicación se ha probado en la red de la Universidad Politécnica de Valencia en transmisiones de secuencias de audio y vídeo entre los campus de Gandía y de Valencia, separados una distancia de unos 70 kilómetros (Fig. 6). Se utilizó un servidor multimedia (Fuente Sincronizadora) ubicado en el campus de Valencia, que obtenía los dos flujos, de forma separada, de un video reproductor profesional, y que los transmitió de forma multicast a 10 receptores localizados en el campus de Gandía. Todos los equipos empleados fueron sincronizados vía un servidor NTP de stratum-1 ubicado en la red nacional académica y de investigación, la Red IRIS.

Dicha transmisión fue evaluada, tanto objetiva como subjetivamente.

Para la sincronización de grupo, al tener todos los receptores las mismas características, se configuró manualmente a uno de ellos como *receptor maestro* y al flujo de audio como el *flujo maestro* ya que los requerimientos en cuanto a sincronización son más estrictos para dicho flujo, comparado con el flujo de vídeo. Para la sincronización inter-flujo los dos procesos de cada reproductor se comunicaban vía *mbus*. El proceso reproductor del flujo esclavo de vídeo adaptó su estado de reproducción según le iba comunicando el proceso del flujo maestro de audio, mediante 'saltos' y 'pausas' en la reproducción de las tramas de vídeo (LDUs) cuando la asincronía detectada superaba un umbral prefijado.

De acuerdo con las conclusiones obtenidas por Steimetz en [16], fijamos los siguientes límites de asincronías permitidas entre flujos:

- ±120 milisegundos como el máximo valor permitido para la asincronía entre receptores para el flujo maestro de audio (para la sincronización de grupo, distribuida)

- ±160 milisegundos (aunque se consideran ideales valores por debajo de ±80 milisegundos) como el máximo valor permitido para la asincronía entre los procesos de reproducción de los flujos de audio y vídeo (sincronización inter-flujo local).

A continuación se presentan los resultados de las dos evaluaciones realizadas.

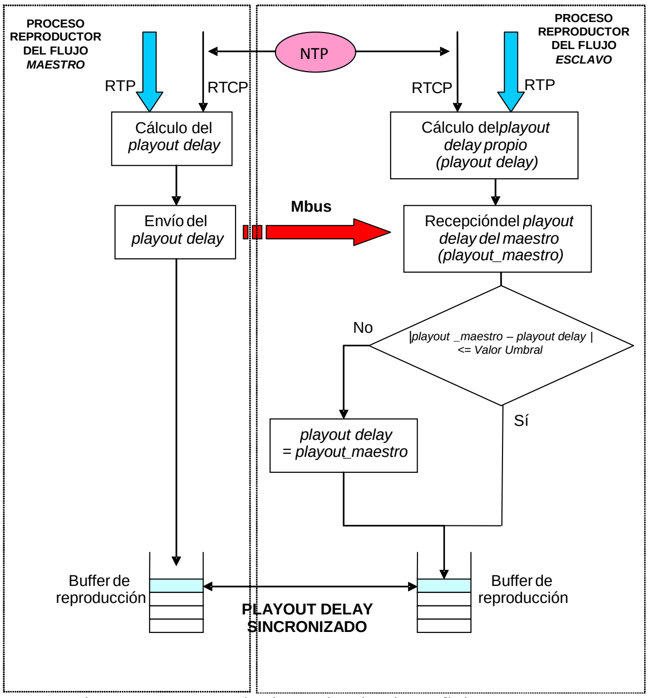


Figura 5. Esquema de sincronización inter-flujo propuesto

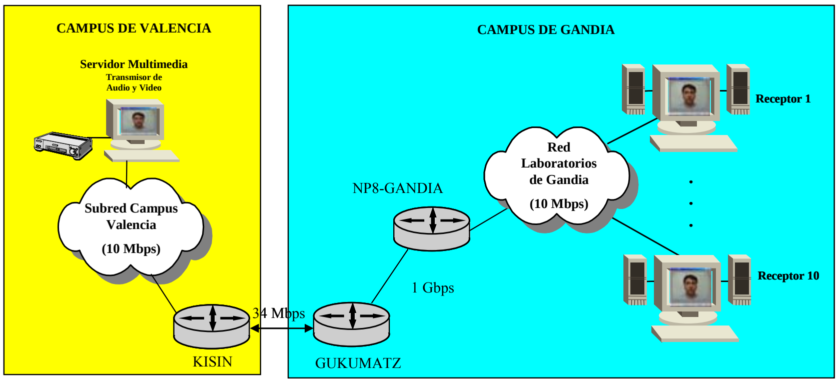


Figura 6. Escenario de prueba

3.1 Resultados de la Evaluación Objetiva

Se probaron las aplicaciones tanto sin activar como activando en las mismas la solución de sincronización propuesta. Sin activarla se comprobó que cada receptor iniciaba la reproducción en diferentes instantes y, además, se consiguió una media de 2,5 segundos de asincronía, inaceptable, en la reproducción del flujo maestro (audio) en los receptores a lo largo de los 10 minutos que duraban las secuencias transmitidas en esta evaluación.

Al activarla se comprobó cómo todos los receptores iniciaban la reproducción de forma sincronizada y continuaban la reproducción de forma sincronizada durante la sesión. La Fig. 7 presenta el valor del *playout delay* (retardo desde el instante de la transmisión de las LDUs) del flujo maestro (audio) en los 10 receptores durante la sesión, cuando se activó la solución presentada en el artículo. Para suavizar las variaciones de las curvas se han representado medias móviles tomando grupos de 100 valores. Se puede apreciar que los *playout delays* en cada receptor se van ajustando al del receptor maestro (línea gruesa), cuyo valor medio está alrededor de 500 milisegundos en la sesión mostrada. El gran incremento inicial del retardo de reproducción es debido al inicio de las aplicaciones durante el cual se produce un alto consumo de recursos de la máquina lo cual aumenta el retardo de procesamiento.

La cantidad de mensajes de control enviados por la Fuente Sincronizadora (paquetes *RTCP APP ACT*) representó solo el 0,14% de la cantidad total de paquetes (de control y de datos) enviados por ésta. Por otro lado, la cantidad de los mensajes de control enviados por los receptores (paquetes *RR EXT*) apenas supuso el 6,88% de la cantidad total de paquetes (de control y de datos) enviados por todas las aplicaciones. También se analizó el valor cuadrático medio de la asincronía de grupo detectada y se observó que en ningún receptor se sobrepasó el límite de 14.400 milisegundos^2 (valor cuadrático del valor máximo permitido, ±120 milisegundos). Los valores obtenidos fueron muy inferiores, obteniendo, por tanto, buenos resultados en la sincronización de grupo.

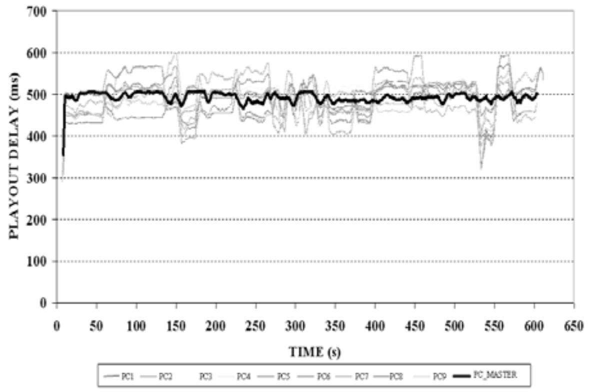


Figura 7. Playout delay del flujo maestro (audio)

Con respecto a la sincronización inter-flujo, también se analizó el valor cuadrático medio de la asincronía entre los procesos reproductores de los flujos de audio y video en cada receptor y se observó que se mantenía la mayor parte del tiempo muy por debajo del valor correspondiente a ±80 milisegundos (6.400 milisegundos^2) y, obviamente, del valor correspondiente a ±160 milisegundos (25.600 milisegundos^2). En la Fig. 8 se muestra la distribución del valor cuadrático medio de la asincronía entre flujos detectada para uno de los receptores (para el resto de receptores los resultados fueron similares). En ella se observa que los límites anteriores (marcados con líneas de puntos) se sobrepasaron en muy pocas ocasiones, en las cuales, la evaluación subjetiva mostró que los efectos ocasionados en la reproducción no fueron demasiado molestos para los usuarios encuestados.

3.2 Resultados de la Evaluación Subjetiva

Tal como se ha indicado, se ha complementado la evaluación objetiva con una evaluación subjetiva realizada a 20 usuarios, ninguno de los cuales tenía experiencia previa en evaluación subjetiva ni en técnicas de sincronización. Se les envió 3 secuencias de 3 minutos de una película de acción con 3 grados de sincronización: sin sincronización alguna, con sólo sincronización inter-flujo y con la sincronización de grupo propuesta (incluyendo inter-flujo). El flujo de vídeo tenía codificación H-261, con 25 tramas/segundo, mientras que el flujo de audio tenía codificación GSM, con 8000 muestras/segundo.

Primero, los usuarios tenían que evaluar la calidad de la sincronización de las secuencias en una escala de 1 a 5 (donde 5 indicaba total sincronización, mientras que 1 indicaba falta de sincronización entre flujos). A continuación, tenían que evaluar la calidad de la presentación también en una escala de 1 a 5 (donde 5 indicaba buena presentación sin efectos anormales –pausas, saltos, chasquidos en el audio, etc.-, mientras que 1 indicaba una presentación muy irritante debido a efectos molestos en la misma) e indicar los efectos apreciados. En ambos casos, un valor de '0' indicaba indecisión del usuario. Ambas escalas se basan en las utilizadas en la recomendación UIT-R BT. 500-11 ([17]).

La Fig. 9 muestra el resultado de la evaluación subjetiva de la calidad de la sincronización. En ella se muestran la valoración media, la máxima y la mínima otorgada por los usuarios a la calidad de la sincronización. Se puede observar cómo la utilización de la propuesta de sincronización de grupo (distribuida y local) obtuvo una buena evaluación, muy parecida a la obtenida con las secuencias con únicamente la sincronización inter-flujo (local), pero adquiriendo en este caso también las sincronización de grupo entre receptores perseguida. La Fig. 10 presenta la degradación de la sincronización percibida por los usuarios en las secuencias mostradas. En las secuencias con sincronización de grupo se detectaron efectos anormales debido a los procesos de sincronización pero fueron descritos como imperceptibles y poco molestos por los usuarios. En la secuencia de la película de acción había cambios frecuentes de planos por lo que las acciones de sincronización (saltos o pausas en la reproducción) resultaban difíciles de apreciar por los usuarios. Además, los usuarios están acostumbrados a ver películas extranjeras donde se ha producido un doblaje en el idioma con lo que ya debido a dicho proceso se pueden observar asincronías entre los flujos de audio y vídeo. Es por ello que entendemos que los usuarios toleraran bien las propias asincronías y las correcciones de las mismas, no considerando dichos efectos como extraños o anormales.

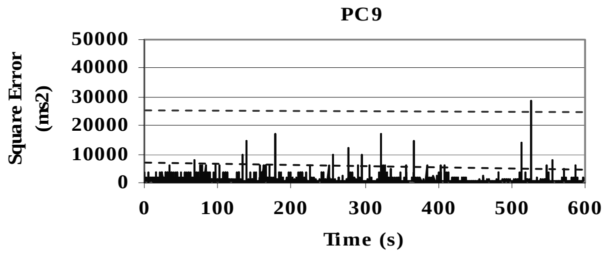


Figura 8. Valor cuadrático medio de la asincronía detectada entre flujos en uno de los receptores (PC9)

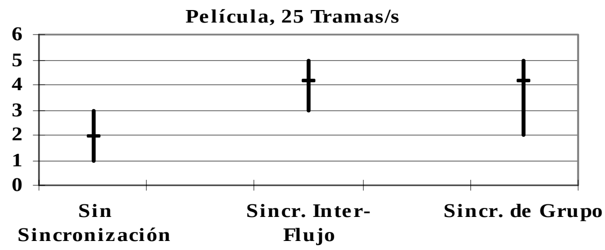


Figura 9. Calidad de la sincronización

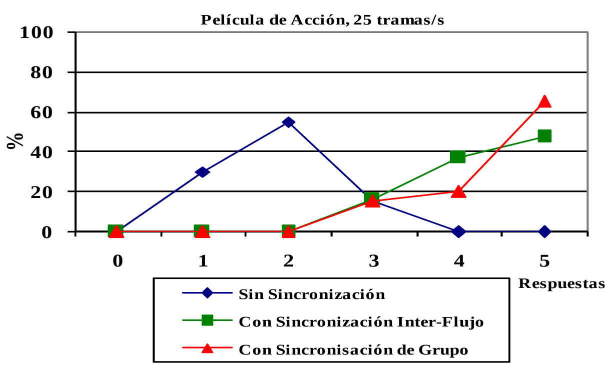


Figura 10. Degradación de la sincronización

4 Conclusiones

En este artículo se ha presentado una posible solución a la problemática de la sincronización de grupo de flujos multimedia. Aprovechando la maleabilidad de los protocolos RTP/RTCP, se propone la modificación de paquetes RTCP y la definición de nuevos paquetes para obtener dicha sincronización de forma fácil y factible.

Dicha solución apenas incrementa la carga de la red y facilita la corrección de las asincronías existentes entre diferentes receptores y entre los flujos en un mismo receptor impidiendo que éstas superen los límites establecidos como aceptables en la literatura relacionada. Al utilizar mensajes RTCP, se consigue mantener una muy baja carga de información de control y mensajes dedicados a la sincronización, en comparación al número total de LDUs transmitidas.

La solución de sincronización de grupo propuesta ha obtenido buenos resultados, tanto en la evaluación objetiva como en la subjetiva, lo cual la valida como una posibilidad a tener en cuenta en la sincronización de grupo multimedia para aplicaciones multimedia distribuidas.

Podemos concluir que nuestra propuesta resultará apropiada para sistemas multimedia distribuidos con varias fuentes y varios receptores, donde se realice una transmisión *multicast* (si las red lo permite) de flujos individuales no multiplexados, a través de una red determinista o con una cierta calidad de servicio garantizada, donde los retardos máximos sean limitados y/o conocidos a priori.

Como trabajo futuro, pretendemos combinar la solución con la posibilidad de que la fuente, si hay problemas de ancho de banda y de acuerdo con la información de realimentación recibida, pueda modificar dinámicamente los parámetros de transmisión (tasa, codificación, etc.) para adaptarse al estado de la red en cada momento y mejorar la calidad del sistema multimedia distribuido. Otra línea futura consiste en estudiar si es necesario enviar la extensión propuesta en todos los paquetes RTCP RR y, en caso de que no sea así, incluir indicaciones desde la fuente para señalar a los receptores cuándo deben enviar la extensión. Esto minimizaría aún más la carga de control introducida por la solución propuesta. Finalmente, nos gustaría implementar nuestra propuesta mediante agentes software para la sincronización multimedia, tal y como se propone en [18].

Referencias

[1] R. Yavatkar, K. Lakshman, “Communication support for distributed collaborative applications”, Multimedia Systems, 2(4), 1994.

[2] I. F. Akyildiz, W. Yen, “Multimedia Group Synchronisation Protocols for Integrated Services Networks”, IEEE JSAC., vol. 14, pp. 162 - 173, Jan. 1996

[3] C. Diot, L. Gautier. “A Distributed Architecture for Multiplayer Interactive Applications on the Internet”, IEEE Network, vol. 13, pp. 6 - 15, Jul./Aug. 1999.

[4] Ishibashi, Y., Hasegawa, T., Tasaka, S., “Group synchronization control for Haptic Media in Networked Virtual Environments”, 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004, HAPTICS ‘04 Proceedings, pp. 106-113.

[5] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. “RTP: A Transport Protocol for Real-Time Applications”, RFC-3550, Jul. 2003.

[7] J. Ott , S. Wenger, N. Sato, C. Burmeister, J. Rey. “Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)”, RFC 4585, Jul. 2006.

[8] C. Burmeister, R. Hakenberg, A. Miyazaki, J. Ott, N. Sato, S. Fukunaga. “Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback: Results of the Timing Rule Simulations”, RFC 4586, , Jul. 2006.

[8] D. L. Mills. "Network Time Protocol", RFC 958, Sept. 1985

[9] F. Boronat, J.C. Guerri, “Analysis and Comparison of Multimedia Inter-stream and Group Synchronisation Algorithms”, IEEE Latin America Transactions, Vol. 3, issue 5, Dic. 2005.

[10] P. V. Rangan, S. Ramanathan, S. Sampathkumar, “Feedback techniques for continuity and synchronization in multimedia information retrieval”, ACM Transactions on Information Systems, Vol. 13, Issue 2, Apr. 1995.

[11] J.C. Guerri. “Especificación y evaluación de prestaciones de un Protocolo de Sinchronización Multimedia Adaptativo con Control de Stream, basado en un Tiempo Global y Técnicas de Realimentación” Tesis doctoral U.P.V. Jun. 1997.

[12] F. Boronat, “Especificación y Evaluación de un algoritmo de Sincronización de Grupo de Flujos Multimedia”, Tesis doctoral U.P.V. abr. 2004.

[13] The Message Bus: http://www.mbus.org.

[14] I. Kouvelas and V. Hardman, "Overcoming workstation scheduling problems in a real-time audio tool", Proc. USENIX, Anaheim, CA, Jan. 1997, pp. 235-242.

[15] S. McCanne and V. Jacobson. “vic: A Flexible Framework for Packet Video”. ACM Multimedia, Nov. 1995, San Francisco, CA, pp. 511-522.

[16] R. Steimetz. “Human Perception of Jitter and Media Skew”, IEEE JSAC, Vol. 14, nº 1, Jan. 1996.

[17] UIT-R BT. 500-11, “Metodología para la evaluación subjetiva de la calidad de las imágenes de televisión”, Jun. 2002.

[18] S. S. Manvi, P. Venkataram, “An agent based synchronization scheme for multimedia applications”, Journal of Systems and Software (JSS), Vol. 79, issue 5, pp. 701-713, May 2006.

Sincronización de grupo multimedia basada en protocolos estándar

F. Boronat, *Member IEEE*, J.C. Guerri, J. Lloret, *Member IEEE y* M. García*, Student Member IEEE*

***Resumen*—La mayoría de las herramientas multimedia actuales utilizan RTP/RTCP para sincronización inter-flujo, pero no para sincronización de grupo. Presentamos una nueva propuesta de modificación de los paquetes RTCP para proporcionar un método basado en la fuente para sincronizar un grupo de receptores y se ha evaluado tanto objetiva como subjetivamente. La solución se aprovecha de la capacidad de los mensajes de realimentación RTCP (paquetes RR) y de la maleabilidad de RTP/RTCP para proporcionar la información necesaria requerida por la solución propuesta de sincronización, definiendo nuevos paquetes APP (paquetes especiales RTCP) útiles para el propósito de la sincronización. Esta modificación apenas incrementa la carga de la red y ayuda a evitar que las asincronías, tanto entre receptores (sincronización distribuida) como entre flujos (sincronización local), sobrepasen los límites fijados en estudios anteriores.**

***Palabras clave*—Comunicaciones Multimedia, Sistemas Multimedia, Protocolos, Sincronización Multimedia**

I. INTRODUCTION

ACTUALMENTE, existen muchas aplicaciones multimedia distribuidas basadas en la cooperación (teleenseñanza, televigilancia, juegos en red, distribución de video con su audio en diferentes idiomas, etc.), las cuales incluyen la transmisión de diferentes flujos (audio, video, texto, datos,…), normalmente de forma multicast, desde una o varias fuentes a uno o varios receptores. Todas ellas incluyen normalmente sincronización intra-flujo (añaden algún mecanismo que garantice las relaciones temporales entre unidades de datos –LDUs o *Logical Data Units*- de un mismo flujo, como, por ejemplo, entre las tramas de una misma secuencia de video) e inter-flujo (garantizando las relaciones temporales entre las LDUs de los diferentes flujos multimedia, como, por ejemplo, la reproducción del audio de un discurso y los movimientos asociados de los labios del locutor del discurso, conocida como sincronización labial o *Lip-Sync*).

Sin embargo, en determinadas aplicaciones se necesita otro tipo de sincronización, denominado *Sincronización de Grupo*, que consiste en garantizar la reproducción sincronizada de todos los flujos tanto localmente (inter-flujo) en cada receptor como, a la vez y globalmente, en todos los receptores (en grupo). Se ocupa de garantizar la reproducción de todos los flujos de forma sincronizada en todos los receptores al mismo tiempo. Se han encontrado muy pocas soluciones incluyendo este tipo de sincronización, entre las que se pueden destacar [1], [2], [3] y [4], todas las cuales se basan en el receptor (*receiver-driven*) y, excepto la presentada en [3] que utiliza RTP/RTCP ([5]), ninguna utiliza protocolos estándar en sus propuestas sino que definen nuevos protocolos con mensajes de control de la sincronización específicos, que se intercambian entre las fuentes y los receptores para obtener la sincronización final deseada. Destacamos la solución presentada por Akyldiz y Yen en [2] y el algoritmo *VTR* (*Virtual Time Rendering*, [4]). Ambas soluciones también se basan en el receptor (*receiver-driven*), utilizan un receptor como referencia para la sincronización (esquema maestro/esclavo) e incluyen intercambio de información entre receptores para sincronizarse con el de referencia, lo cual implica una carga de red considerable. El algoritmo propuesto en [2] también propone un mecanismo para sincronizar el instante inicial de la reproducción en todos los receptores.

Por otro lado, también se han encontrado dos RFCs, la 4585 ([6]) y la 4586 ([7]) que definen nuevas extensiones para el perfil *Audio-visual Profile (AVP) for RTCP-based feedback (RTP/AVPF)*, que permite a los receptores proporcionar realimentación de forma más inmediata a las fuentes y así permitir una adaptación de la transmisión a corto plazo y la posibilidad de implementar mecanismos de recuperación. La RFC 4585 ([6]) también define un pequeño grupo de mensajes de realimentación RTCP de propósito general. Tal como se explica más adelante, en nuestra solución se han definido nuevas extensiones para determinados paquetes RTCP y, además, nuevos mensajes RTCP para realimentación útiles para el propósito de la sincronización de grupo deseada.

Se presenta un método novedoso para obtener la sincronización de grupo, basado en RTP/RTCP ([5]) y en NTP ([8]), minimizando el tráfico de control con respecto a las soluciones anteriores e incluyendo las técnicas más comunes de sincronización utilizadas por los algoritmos y soluciones más populares. En [9] se detallan dichas técnicas y se compara nuestra propuesta con dichas soluciones.

A continuación, en la sección 2 se expone la solución propuesta. En la sección 3 se muestran los resultados de los dos tipos de evaluación realizadas, finalizando el artículo con las conclusiones del mismo y las referencias bibliográficas.

Fernando Boronat Seguí, Juan Carlos Guerri Cebollada y Jaime Lloret Mauri son profesores del Departamento de Comunicaciones de la Universidad Politécnica de Valencia - Escuela Politécnica Superior de Gandia, en la Ctra. Nazaret-Oliva S/N, 46730 Grao de Gandia (VALENCIA) (e-mail: {fboronat, jcguerri, jlloret}@dcom.upv.es)

Miguel García Pineda es estudiante colaborador en el Departamento de Comunicaciones de la UPV

II. PROPUESTA DE SINCRONIZACIÓN

La solución presentada será de aplicación en escenarios con sistemas distribuidos con una o varias fuentes de flujos multimedia transmitiendo, de forma multicast, y uno o varios receptores de dichos flujos, utilizando redes de comunicaciones determinísticas con unos requerimientos mínimos de calidad de servicio (al menos, deberá ser conocido o acotado el retardo extremo a extremo de la red). La estructura de la propuesta, en cuanto a funcionalidad, está basada en el protocolo *Feedback* ([10]), pero añadiendo la utilización de un tiempo global proporcionado por el protocolo NTP, tal y como se propone en el protocolo *Feedback Global* ([11]). En [10] se trabaja con relojes locales. Las soluciones propuestas en [10] y [11] sólo incluyen técnicas de sincronización intra e inter-flujo, son adaptativas, válidas para multicast, utilizan esquemas maestro/esclavo y técnicas de realimentación para intercambiar información entre fuentes y receptores.

Para resolver el problema de la sincronización en los receptores, dividimos el proceso en dos fases (Fig. 1):

1. Conseguir que todos los receptores inicien la reproducción de uno de los flujos, considerado como *flujo maestro*, en el mismo instante (*Instante Inicial de Reproducción*) y que, a partir de dicho instante, continúen la reproducción de dicho flujo de forma sincronizada (llamaremos a este proceso *sincronización distribuida de grupo entre receptores*).

2. Conseguir que localmente, en cada receptor, se reproduzcan de forma sincronizada todos los flujos que deba reproducir dicho receptor (*sincronización local inter-flujo*).

Para ello, nuestra propuesta se basa en dos *esquemas maestro/esclavo*. Por un lado, existirá un *receptor maestro* que servirá de referencia para la sincronización de grupo, entre receptores, y, por otro lado, existirá un *flujo maestro* que servirá de referencia para la sincronización inter-flujo interna en cada receptor.

En la Fig. 1 se puede apreciar la existencia de una transmisión, que puede ser *multicast* o *unicast*, de flujos multimedia mediante RTP desde una o varias fuentes transmisoras a uno o varios receptores. Uno de los flujos multimedia es tomado como flujo *maestro* (líneas y flechas de mayor grosor) y, además, de entre todos los receptores se selecciona uno de ellos como *receptor maestro* (gris en la figura), cuyo estado de reproducción del flujo *maestro* será tomado como referencia para determinar el estado de reproducción de cada uno de los demás receptores (*esclavos*). Este *receptor maestro* podrá ser elegido de varias maneras, según determinados criterios (tal y como se describe en [12]). Se utilizará RTCP para enviar mensajes de control durante la sesión.

La fuente transmisora del flujo maestro se convertirá en la *Fuente Sincronizadora* y será la que controlará que la reproducción de los receptores se haga de la forma más sincronizada posible, debiendo procesar y analizar la información de realimentación que estos le enviarán de forma, más o menos, periódica.

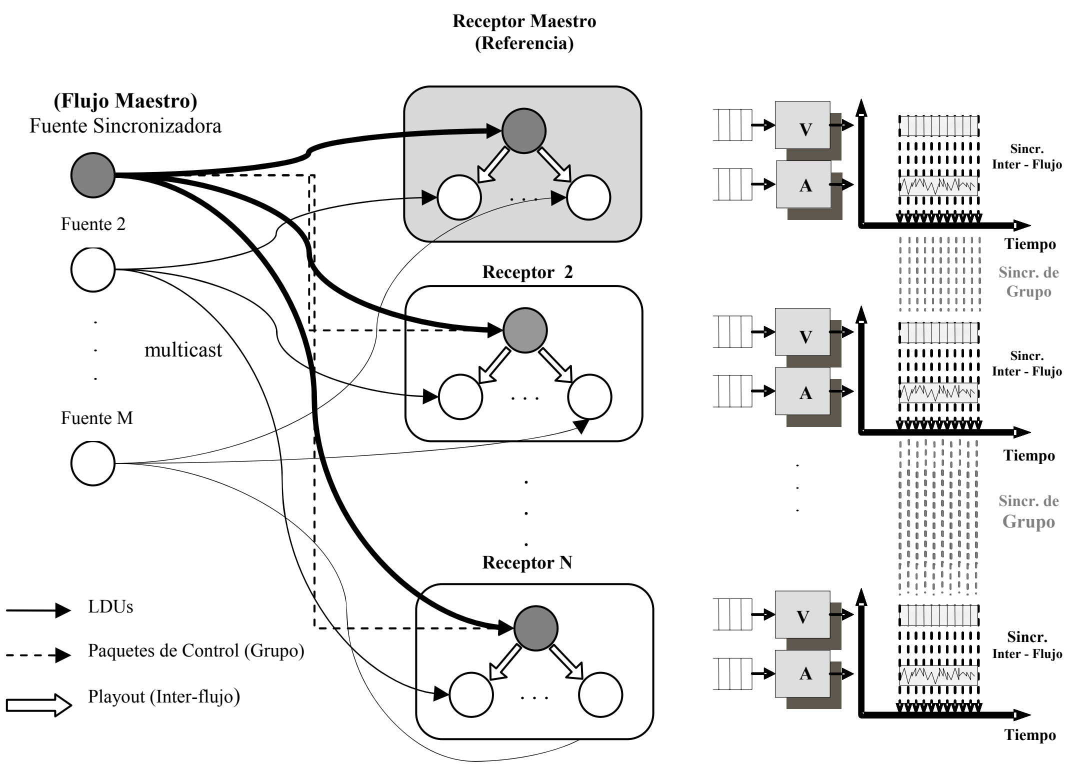


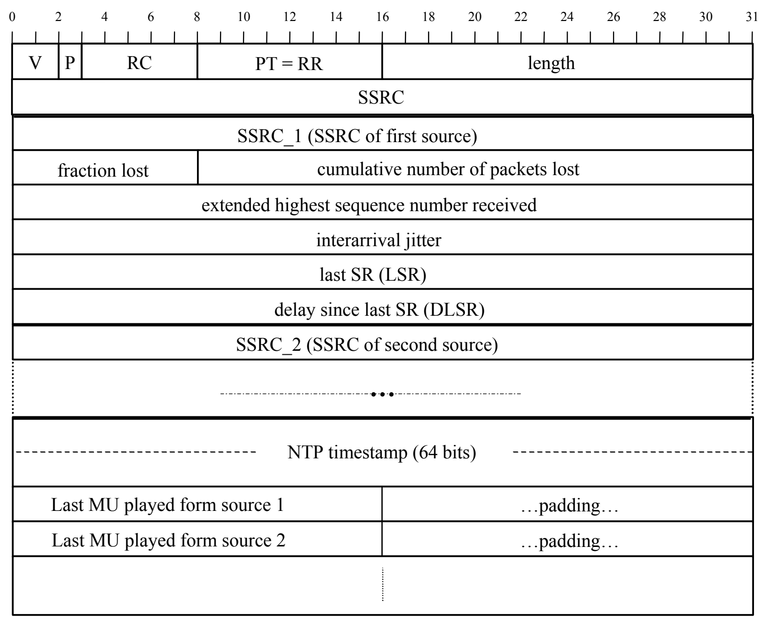
Fig. 1. Sincronización Inter-flujo y de grupo

Los paquetes de realimentación en RTCP son los denominados *RTCP Receiver Reports* (paquetes RTCP RR, [5]) pero la información que contienen no es suficiente para nuestro propósito final de la sincronización. Es por ello que proponemos la modificación de dichos paquetes para incluir la información necesaria para dicho propósito. Además, se definirán nuevos paquetes RTCP APP (*Application-defined RTCP packet*, [5]) que utilizará la fuente para indicar cuándo se debe iniciar la reproducción y también las posteriores correcciones a los receptores en sus procesos de reproducción, cuando detecte que están entrando en situaciones de asincronía (paquetes que denominaremos '*paquetes de acción*').

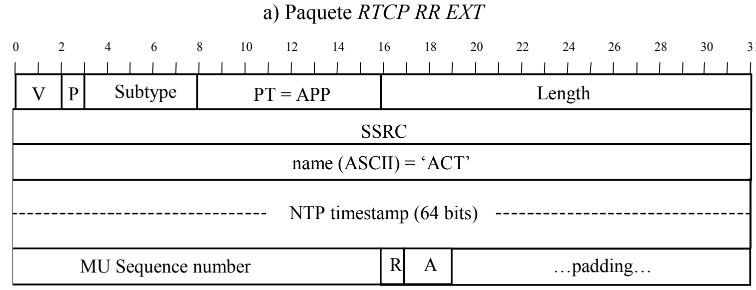
Para definir los *paquetes de acción* proponemos el uso de nuevos paquetes de control RTCP APP ([5]), que hemos denominado *paquetes RTCP APP ACT* (de '*acción*'), con una extensión dependiente de nuestra aplicación, incluyendo un número de secuencia de LDU y la marca de tiempo, en unidades NTP, del instante en que la LDU con dicho número de secuencia deberá ser reproducida por todos los receptores (Fig. 2b). Este paquete también servirá para indicar el instante de inicio común de la reproducción a todos los receptores de la primera LDU del flujo maestro.

El funcionamiento general del algoritmo propuesto es el mostrado en la Fig. 3, donde se representa la fuente sincronizadora (transmisora del flujo *maestro*) y los receptores *i* y *j* de la sesión.

Durante la sesión, la fuente sincronizadora irá recibiendo, de uno en uno, los paquetes *RTCP RR EXT* pertenecientes a todos los receptores que estén reproduciendo el flujo maestro transmitido por ella. De dichos paquetes extraerá la información relacionada con el identificador del receptor (identificador *SSRC*, definido en [5]), la última LDU reproducida por el mismo y el instante NTP en que dicho receptor reprodujo dicha LDU. Esta información se irá guardando en una tabla creada por la propia fuente sincronizadora con un número de registros igual al número de receptores participantes en la sesión (*n*), con la estructura mostrada en la tabla 1. En los casos en que la fuente reciba un segundo paquete *RTCP RR EXT* procedente de un mismo receptor antes de completar toda la tabla, actualizará la información, con el fin de mantener la tabla con los valores más recientes.



a) Paquete *RTCP RR EXT*



b) Paquete *RTCP APP ACT*

Fig. 2. Formato de los paquetes propuestos

La columna "*bit de reproducción*" (R_i) indica si el receptor está o no activo y se utiliza para saber si el receptor incluido en la sesión está o no reproduciendo el flujo maestro y, por tanto, su información (LDU_i y NTP_i) deberá ser tomada en cuenta (bit a '1') o no (bit a '0') para realizar el cálculo del punto de reproducción de referencia. Este bit será necesario para poder considerar los abandonos de los receptores durante la sesión y evitar que los datos referentes a receptores no presentes en la sesión afecten al resto en un momento dado.

Una vez completada la tabla con los nuevos datos procedentes de todos los receptores activos, se estará en disposición de elegir a uno de los receptores como referencia siguiendo algún criterio específico (por ejemplo, el receptor más lento en su reproducción, o el más rápido, etc.).

Lo ideal, a la hora de calcular la referencia o receptor *maestro* con el cual se sincronizarán todos los demás receptores, sería que todos los receptores enviaran un paquete de control con dicha información a la vez, es decir, con la misma referencia temporal o instante NTP. Esto, lógicamente, en sesiones con un elevado número de usuarios podría suponer un envío masivo de paquetes de todos los receptores a la fuente en ciertos instantes, lo cual podría colapsarlo, afectando a la escalabilidad de la solución propuesta. Este ha sido uno de los motivos por los que se ha elegido el paquete RTCP RR para enviar la información necesaria descrita anteriormente. Tal y como describe la RFC 1889 ([5], en el Anexo 1, apartado 6.2, *Intervalo de transmisión RTCP*), cada receptor enviará su paquete de informe *RTCP RR EXT* de forma aleatoria. Por lo tanto, el momento NTP con el que los receptores enviarán sus paquetes *RTCP RR EXT* no será el mismo. Debido a esta aleatoriedad en el envío, la fuente se verá obligada a buscar una relación entre la última LDU consumida y el tiempo global y 'real' NTP. Esto es posible gracias a las marcas de tiempo NTP y RTP que contienen los paquetes RTCP.

Una vez conseguida la sincronización de grupo, es decir, cuando ya todos los receptores estén reproduciendo el flujo maestro de forma sincronizada, también será necesario un mecanismo adicional para conseguir que, localmente en cada receptor, los flujos que se reproduzcan en el mismo también lo hagan de forma sincronizada entre ellos (sincronización inter-flujo local). Para ello se hará uso de un bus interno de comunicación entre los procesos de reproducción del receptor, denominado *mbus* (cuya especificación está en [13]).

Mediante mensajes a través de *mbus* el proceso de reproducción del flujo maestro envía su estado de reproducción a todos los demás procesos reproductores de los flujos esclavos del receptor (Fig. 4) para que estos se adapten a dicho estado, mediante 'saltos' (o, lo que es lo mismo, descarte de las LDUs del buffer cuyo instante de reproducción ya haya pasado) y/o 'pausas' (lo que equivale a repetir la reproducción de la última LDU hasta que se deba reproducir la siguiente almacenada en el buffer de reproducción) en su reproducción.

TABLA I
INFORMACIÓN MANEJADA POR LA FUENTE

SSRC	**Última LDU**	**NTP timestamp**	**Bit de Reproducción R_i**
$SSRC_1$	LDU_1	NTP_1	bit_1
$SSRC_2$	LDU_2	NTP_2	bit_2
$SSRC_n$	LDU_n	NTP_n	bit_n

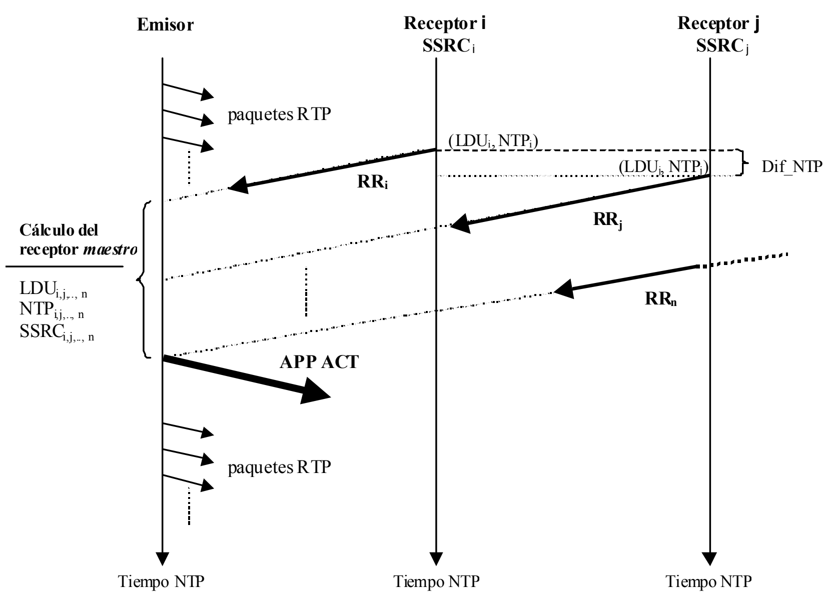


Fig. 3. Funcionamiento General

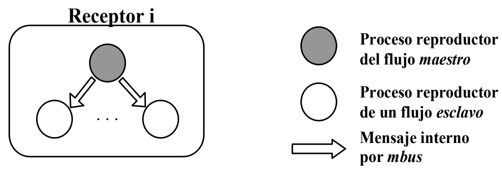


Fig. 4. Sincronización local inter-flujo a través del bus interno *mbus*

En la Fig. 5 aparece el diagrama de flujos del intercambio de información entre los procesos del proceso reproductor del flujo maestro y el proceso reproductor de uno de los flujos esclavos. El proceso del flujo maestro le comunica al del flujo esclavo el valor de su *playout delay* en cada momento. Se trata del retardo de reproducción de la LDU que está reproduciendo en dicho instante, esto es, el retardo transcurrido desde que se transmitió dicha LDU desde la Fuente Sincronizadora hasta que es reproducida. Para evitar continuas adaptaciones, el proceso del flujo esclavo lo compara con el suyo propio y sólo hace correcciones si la diferencia entre los dos valores es superior a un determinado umbral que se configurará según las aplicaciones.

Ya que cada proceso reproductor de un flujo esclavo no tiene la misma referencia de reloj que el del flujo maestro, se hace uso de las marcas de tiempo NTP y del '*mapeado'* entre marcas RTP y marcas NTP para poder obtener una referencia común y así poder realizar la comparación de los valores del *playout delay.*

III. Evaluación

La propuesta ha sido implementada en una aplicación con dos flujos, uno de audio y otro de video, formada por herramientas Mbone, basadas en RTP, modificadas, como son *RAT* ([14]), para transmisión multicast del flujo de audio, y *VIC* ([15]), para la transmisión multicast del flujo de vídeo. Dicha aplicación se ha probado en la red de la Universidad Politécnica de Valencia en transmisiones de secuencias de audio y vídeo entre los campus de Gandía y de Valencia, separados una distancia de unos 70 kilómetros (Fig. 6). Se utilizó un servidor multimedia (Fuente Sincronizadora) ubicado en el campus de Valencia, que obtenía los dos flujos, de forma separada, de un video reproductor profesional, y que los transmitió de forma multicast a 10 receptores localizados en el campus de Gandía. Todos los equipos empleados fueron sincronizados vía un servidor NTP de stratum-1 ubicado en la red nacional académica y de investigación, la Red IRIS.

Dicha transmisión fue evaluada, tanto objetiva como subjetivamente.

Para la sincronización de grupo, al tener todos los receptores las mismas características, se configuró manualmente a uno de ellos como *receptor maestro* y al flujo de audio como el *flujo maestro* ya que los requerimientos en cuanto a sincronización son más estrictos para dicho flujo, comparado con el flujo de vídeo. Para la sincronización inter-flujo los dos procesos de cada reproductor se comunicaban vía *mbus*. El proceso reproductor del flujo esclavo de vídeo adaptó su estado de reproducción según le iba comunicando el proceso del flujo maestro de audio, mediante 'saltos' y 'pausas' en la reproducción de las tramas de vídeo (LDUs) cuando la asincronía detectada superaba un umbral prefijado.

De acuerdo con las conclusiones obtenidas por Steimetz en [16], fijamos los siguientes límites de asincronías permitidas entre flujos:

- ±120 milisegundos como el máximo valor permitido para la asincronía entre receptores para el flujo maestro de audio (para la sincronización de grupo, distribuida)
- ±160 milisegundos (aunque se consideran ideales valores por debajo de ±80 milisegundos) como el máximo valor permitido para la asincronía entre los procesos de reproducción de los flujos de audio y vídeo (sincronización inter-flujo local).

A continuación se presentan los resultados de las dos evaluaciones realizadas.

A. Resultados de la Evaluación Objetiva

Se probaron las aplicaciones tanto sin activar como activando en las mismas la solución de sincronización propuesta. Sin activarla se comprobó que cada receptor iniciaba la reproducción en diferentes instantes y, además, se consiguió una media de 2,5 segundos de asincronía, inaceptable, en la reproducción del flujo maestro (audio) en los receptores a lo largo de los 10 minutos que duraban las secuencias transmitidas en esta evaluación.

Al activarla se comprobó cómo todos los receptores iniciaban la reproducción de forma sincronizada y continuaban la reproducción de forma sincronizada durante la sesión. La Fig. 7 presenta el valor del *playout delay* (retardo desde el instante de la transmisión de las LDUs) del flujo maestro (audio) en los 10 receptores durante la sesión, cuando se activó la solución presentada en el artículo. Para suavizar las variaciones de las curvas se han representado medias móviles tomando grupos de 100 valores. Se puede apreciar que los *playout delays* en cada receptor se van ajustando al del receptor maestro (línea gruesa), cuyo valor medio está alrededor de 500 milisegundos en la sesión mostrada. El gran incremento inicial del retardo de reproducción es debido al inicio de las aplicaciones durante el cual se produce un alto consumo de recursos de la máquina lo cual aumenta el retardo de procesamiento.

La cantidad de mensajes de control enviados por la Fuente Sincronizadora (paquetes *RTCP APP ACT*) representó solo el 0,14% de la cantidad total de paquetes (de control y de datos) enviados por ésta. Por otro lado, la cantidad de los mensajes de control enviados por los receptores (paquetes *RR EXT*) apenas supuso el 6,88% de la cantidad total de paquetes (de control y de datos) enviados por todas las aplicaciones. También se analizó el valor cuadrático medio de la asincronía de grupo detectada y se observó que en ningún receptor se sobrepasó el límite de 14.400 milisegundos^2 (valor cuadrático del valor máximo permitido, ±120 milisegundos). Los valores obtenidos fueron muy inferiores, obteniendo, por tanto, buenos resultados en la sincronización de grupo.

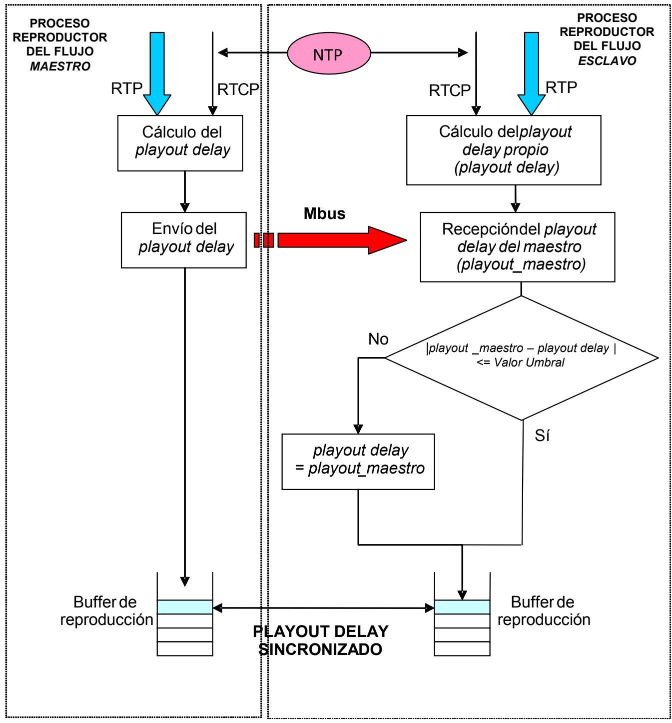


Fig. 5. Esquema de sincronización inter-flujo propuesto

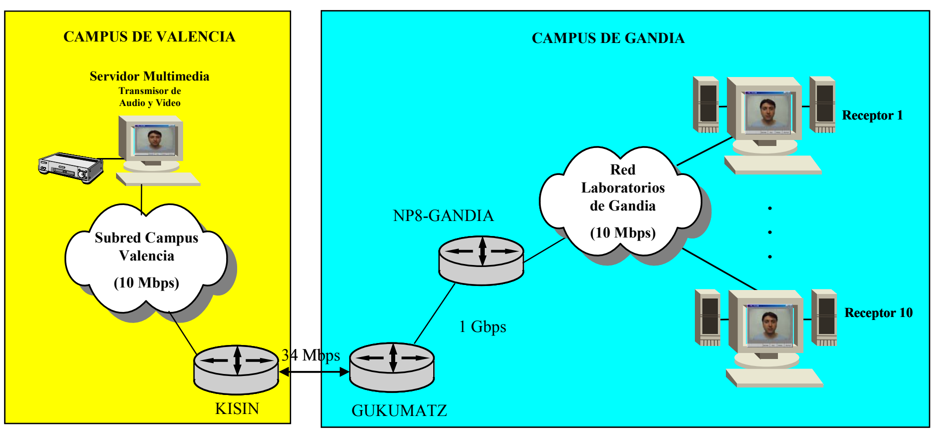


Fig. 6. Escenario de prueba

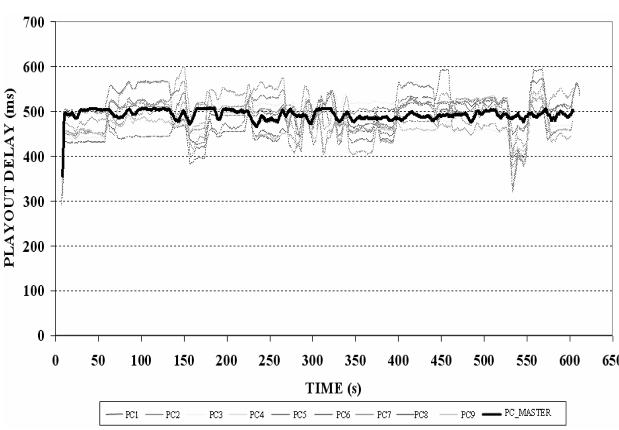


Figura 7. Playout delay del flujo maestro (audio)

Con respecto a la sincronización inter-flujo, también se analizó el valor cuadrático medio de la asincronía entre los procesos reproductores de los flujos de audio y video en cada receptor y se observó que se mantenía la mayor parte del tiempo muy por debajo del valor correspondiente a ±80 milisegundos (6.400 milisegundos^2) y, obviamente, del valor correspondiente a ±160 milisegundos (25.600 milisegundos^2). En la Fig. 8 se muestra la distribución del valor cuadrático medio de la asincronía entre flujos detectada para uno de los receptores (para el resto de receptores los resultados fueron similares). En ella se observa que los límites anteriores (marcados con líneas de puntos) se sobrepasaron en muy pocas ocasiones, en las cuales, la evaluación subjetiva mostró que los efectos ocasionados en la reproducción no fueron demasiado molestos para los usuarios encuestados.

B. Resultados de la Evaluación Subjetiva

Tal como se ha indicado, se ha complementado la evaluación objetiva con una evaluación subjetiva realizada a 20 usuarios, ninguno de los cuales tenía experiencia previa en evaluación subjetiva ni en técnicas de sincronización. Se les envió 3 secuencias de 3 minutos de una película de acción con 3 grados de sincronización: sin sincronización alguna, con sólo sincronización inter-flujo y con la sincronización de grupo propuesta (incluyendo inter-flujo). El flujo de vídeo tenía codificación H-261, con 25 tramas/segundo, mientras que el flujo de audio tenía codificación GSM, con 8000 muestras/segundo.

Primero, los usuarios tenían que evaluar la calidad de la sincronización de las secuencias en una escala de 1 a 5 (donde 5 indicaba total sincronización, mientras que 1 indicaba falta de sincronización entre flujos). A continuación, tenían que evaluar la calidad de la presentación también en una escala de 1 a 5 (donde 5 indicaba buena presentación sin efectos anormales –pausas, saltos, chasquidos en el audio, etc.-, mientras que 1 indicaba una presentación muy irritante debido a efectos molestos en la misma) e indicar los efectos apreciados. En ambos casos, un valor de ‘0’ indicaba indecisión del usuario. Ambas escalas se basan en las utilizadas en la recomendación UIT-R BT. 500-11 ([17]).

La Fig. 9 muestra el resultado de la evaluación subjetiva de la calidad de la sincronización. En ella se muestran la valoración media, la máxima y la mínima otorgada por los usuarios a la calidad de la sincronización. Se puede observar cómo la utilización de la propuesta de sincronización de grupo (distribuida y local) obtuvo una buena evaluación, muy parecida a la obtenida con las secuencias con únicamente la sincronización inter-flujo (local), pero adquiriendo en este caso también las sincronización de grupo entre receptores perseguida. La Fig. 10 presenta la degradación de la sincronización percibida por los usuarios en las secuencias mostradas. En las secuencias con sincronización de grupo se detectaron efectos anormales debido a los procesos de sincronización pero fueron descritos como imperceptibles y poco molestos por los usuarios. En la secuencia de la película de acción había cambios frecuentes de planos por lo que las acciones de sincronización (saltos o pausas en la reproducción) resultaban difíciles de apreciar por los usuarios. Además, los usuarios están acostumbrados a ver películas extranjeras donde se ha producido un doblaje en el idioma con lo que ya debido a dicho proceso se pueden observar asincronías entre los flujos de audio y vídeo. Es por ello que entendemos que los usuarios toleraran bien las propias asincronías y las correcciones de las mismas, no considerando dichos efectos como extraños o anormales.

En este artículo se ha presentado una posible solución a la problemática de la sincronización de grupo de flujos multimedia. Aprovechando la maleabilidad de los protocolos RTP/RTCP, se propone la modificación de paquetes RTCP y la definición de nuevos paquetes para obtener dicha sincronización de forma fácil y factible.

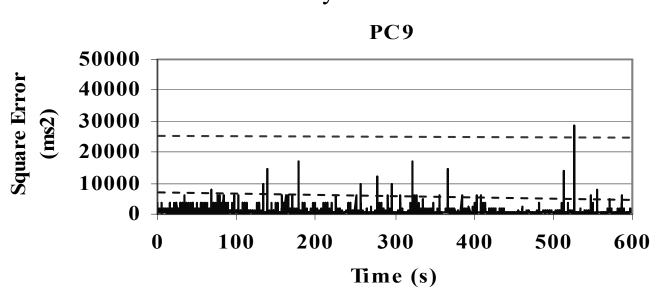


Fig. 8. Valor cuadrático medio (en ms^2) de la asincronía detectada entre flujos en uno de los receptores (PC9)

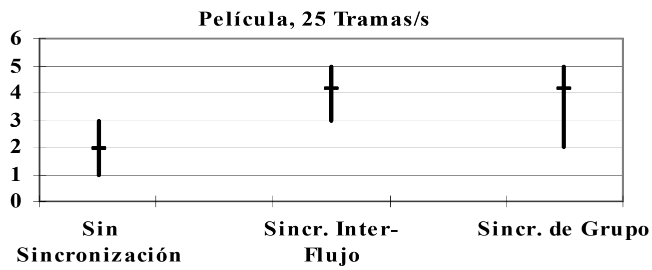


Fig. 9. Calidad de la sincronización

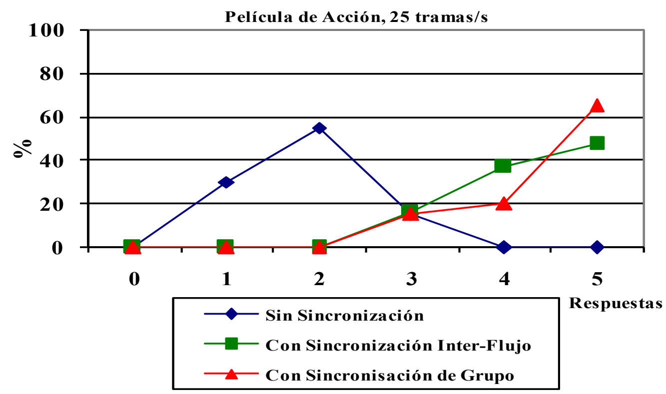


Fig. 10. Degradación de la sincronización

IV. CONCLUSIONES

Dicha solución apenas incrementa la carga de la red y facilita la corrección de las asincronías existentes entre diferentes receptores y entre los flujos en un mismo receptor impidiendo que éstas superen los límites establecidos como aceptables en la literatura relacionada. Al utilizar mensajes RTCP, se consigue mantener una muy baja carga de información de control y mensajes dedicados a la sincronización, en comparación al número total de LDUs transmitidas.

La solución de sincronización de grupo propuesta ha obtenido buenos resultados, tanto en la evaluación objetiva como en la subjetiva, lo cual la valida como una posibilidad a tener en cuenta en la sincronización de grupo multimedia para aplicaciones multimedia distribuidas.

Podemos concluir que nuestra propuesta resultará apropiada para sistemas multimedia distribuidos con varias fuentes y varios receptores, donde se realice una transmisión *multicast* (si las red lo permite) de flujos individuales no multiplexados, a través de una red determinista o con una cierta calidad de servicio garantizada, donde los retardos máximos sean limitados y/o conocidos a priori.

Como trabajo futuro, pretendemos combinar la solución con la posibilidad de que la fuente, si hay problemas de ancho de banda y de acuerdo con la información de realimentación recibida, pueda modificar dinámicamente los parámetros de transmisión (tasa, codificación, etc.) para adaptarse al estado de la red en cada momento y mejorar la calidad del sistema multimedia distribuido. Otra línea futura consiste en estudiar si es necesario enviar la extensión propuesta en todos los paquetes RTCP RR y, en caso de que no sea así, incluir indicaciones desde la fuente para señalar a los receptores cuándo deben enviar la extensión. Esto minimizaría aún más la carga de control introducida por la solución propuesta. Finalmente, nos gustaría implementar nuestra propuesta mediante agentes software para la sincronización multimedia, tal y como se propone en [18].

V. REFERENCIAS

[1] R. Yavatkar, K. Lakshman, "Communication support for distributed collaborative applications", Multimedia Systems, 2(4), 1994.

[2] I. F. Akyildiz, W. Yen, "Multimedia Group Synchronisation Protocols for Integrated Services Networks", *IEEE JSAC*., vol. 14, pp. 162 - 173, Jan. 1996

[3] C. Diot, L. Gautier. "A Distributed Architecture for Multiplayer Interactive Applications on the Internet", *IEEE Network*, vol. 13, pp. 6 - 15, Jul./Aug. 1999.

[4] Ishibashi, Y., Hasegawa, T., Tasaka, S., "Group synchronization control for Haptic Media in Networked Virtual Environments", *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2004, HAPTICS '04 Proceedings, pp. 106-113.

[5] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. "RTP: A Transport Protocol for Real-Time Applications", RFC-3550, Jul. 2003.

[6] J. Ott , S. Wenger, N. Sato, C. Burmeister, J. Rey. "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, Jul. 2006.

[7] C. Burmeister, R. Hakenberg, A. Miyazaki, J. Ott, N. Sato, S. Fukunaga. "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback: Results of the Timing Rule Simulations", RFC 4586, , Jul. 2006.

[8] D. L. Mills. "Network Time Protocol", RFC 958, Sept. 1985

[9] F. Boronat, J.C. Guerri, "Analysis and Comparison of Multimedia Inter-stream and Group Synchronisation Algorithms", *IEEE Latin America Transactions*, Vol. 3, issue 5, Dic. 2005.

[10] P. V. Rangan, S. Ramanathan, S. Sampathkumar, "Feedback techniques for continuity and synchronization in multimedia information retrieval", *ACM Transactions on Information Systems*, Vol. 13, Issue 2, Apr. 1995.

[11] J.C. Guerri. "Especificación y evaluación de prestaciones de un Protocolo de Sinchronización Multimedia Adaptativo con Control de Stream, basado en un Tiempo Global y Técnicas de Realimentación" Tesis doctoral U.P.V. Jun. 1997.

[12] F. Boronat, "Especificación y Evaluación de un algoritmo de Sincronización de Grupo de Flujos Multimedia", Tesis doctoral U.P.V. abr. 2004.

[13] The Message Bus: http://www.mbus.org.

[14] I. Kouvelas and V. Hardman, "Overcoming workstation scheduling problems in a real-time audio tool", *Proc. USENIX*, Anaheim, CA, Jan. 1997, pp. 235-242.

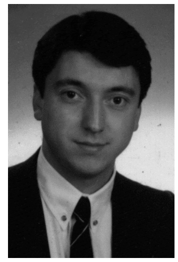
[15] S. McCanne and V. Jacobson. "VIC: A Flexible Framework for Packet Video". *ACM Multimedia*, Nov. 1995, San Francisco, CA, pp. 511-522.

[16] R. Steimetz. "Human Perception of Jitter and Media Skew", *IEEE JSAC*, Vol. 14, nº 1, Jan. 1996.

[17] UIT-R BT. 500-11, "Metodología para la evaluación subjetiva de la calidad de las imágenes de televisión", Jun. 2002.

[18] S. S. Manvi, P. Venkataram, "An agent based synchronization scheme for multimedia applications", *Journal of Systems and Software* (JSS), Vol. 79, issue 5, pp. 701-713, May 2006.

VI. BIOGRAFÍAS



Fernando Boronat Seguí nació en Gandia (España) y estudió en la Universidad Politécnica de Valencia (UPV) en España, donde recibió el título de ingeniero de telecomunicación en 1993. Obtuvo el título de doctor en 2004 en la UPV.

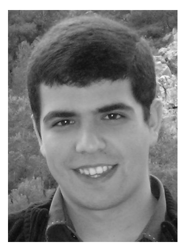
A partir de 1994 trabajó para varias empresas de telecomunicación antes de volver a la UPV en 1996 donde es profesor del Departamento de Comunicaciones en la Escuela Politécnica Superior de Gandia (EPSG). Sus tópicos de interés son las redes de comunicaciones, los sistemas y las aplicaciones multimedia y los protocolos de sincronización multimedia. Es miembro del IEEE desde 1993 y de varios comités de congresos nacionales e internacionales, así como revisor de congresos y revistas internacionales.

Juan Carlos Guerri Cebollada obtuvo el grado de doctor en 1997 en la UPV.

Es profesor en la UPV y responsable del grupo de investigación de Comunicaciones Multimedia, incluido en el Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM) de la UPV. Es miembro de varios comités de conferencias nacionales e internacionales.

Jaime Lloret Mauri recibió el título de licenciado en ciencias físicas en 1997, el título de ingeniero electrónico en la Universidad de Valencia (España) y el título de doctor en comunicaciones de la UPV en 2006.

Es instructor profesional certificado por Cisco y también imparte clases en la EPSG en la UPV. Ha trabajado como administrador de redes en diversas empresas. Actualmente sus temas de interés son las redes P2P y las redes de sensores. Es miembro del IEEE y de IASTED, y forma parte de varios comités de conferencias nacionales e internacionales.



Miguel García Pineda es ingeniero técnico de Telecomunicación desde 2005 por la UPV.

Actualmente está finalizando los estudios de ingeniería de telecomunicación en la UPV y es miembro estudiante del IEEE. En la actualidad está investigando en redes de sensores, redes ad-hoc y redes P2P.

A new neighbour selection strategy for group-based wireless sensor networks

Miguel Garcia[1], Diana Bri[2], Fernando Boronat[3], Jaime Lloret[4]
Department of Communications, Polytechnic University of Valencia
[1,2]{migarpi, diabrmo}@ posgrado.upv.es, [3,4]{ fboronat, jlloret}@dcom.upv.es

Abstract

In any type of networks a neighbour selection method is needed to form the topology of the network and to know which node the information has to be sent to reach a destination. Nowadays, several selection strategies exist that are based on different aspects and mainly designed to work in common networks. In this paper we will show our study about those different methods and, then we show the development of a suitable neighbour selection strategy for group-based wireless sensor networks (WSN) that is based on a capacity parameter defined by us and the new neighbour distance. We also present the proposal architecture for WSNs and the protocol when a new node joins a group and has to select its neighbours.

1. Introduction

The number of nodes, the number of connections in the network, the degree of the nodes and the diameter of the network are main parameters used to determine a network topology [1], but there are others, such as the bandwidth, that could be considered.

In order to obtain a network topology, its nodes have to be interconnected, so there has to be a strategy that nodes must follow to choose their neighbours. A node has many ways to choose its neighbours. Each topology, each system and each protocol has the development and the improvement of its neighbour selection as its main goal. Examples given of neighbour discovery mechanisms are reference [2] in pure P2P networks, references [3] and [4] in hybrid P2P networks, reference [5] in unstructured P2P networks, reference [6] in content delivery systems and reference [7] in distributing systems. Both the routing protocols and the interconnection strategy are the main issues taken into account in all type of data networks.

Although grouping nodes give many benefits to the network, we have found very few works about it, but none of them are for WSN and none of them tackles the way of establishing connections and discovering neighbours between nodes from different groups.

The structure of the paper is as follows. The related works are presented in the next Section. Section 3 describes the group-based architecture and the neighbour selection proposal. Finally, in section 4, we conclude the paper giving the benefits of our proposal compared with the other neighbour selection systems.

2. Related works

A neighbour selection algorithm decides which parameters are used, and their values, to select the best neighbour node. This election is given taking into account that it has to accomplish an objective function. It must take also into account how connections are distributed in the network.

Several neighbour selection algorithms exist. Some basic algorithms that depend on the node upstream bandwidth (BW) or on the number of connections (n) are the following ones [2]:

i) Greedy: This algorithm selects the node with the highest value of the neighbour selection parameter (NSP) among all candidates. This parameter is given by expression 1.

$$NSP = \frac{BW}{n+1} \tag{1}$$

ii) Fit: Let b_0 be the downlink bandwidth of the requester. If all neighbouring candidates have a value NSP<b_0, it selects the node with the highest NSP value, otherwise, chooses the one that has minimal positive value of NSP-b_0.

iii) Fastest Link: This algorithm selects the node with highest upstream bandwidth without taking into account the number of connections.

iv) Random: It chooses its neighbour randomly despite its neighbour upstream bandwidth and their number of connections.

Throughout the years more complex types of strategies for neighbour selection have been developed.

DOI 10.1109/ICNS.2008.18

2.1. Based on genetic algorithms

A genetic algorithm (GA) is stochastic and an adaptive heuristic search technique used in computing to find exact or approximate solutions to optimization and search problems which is based on the mechanism of natural selection, genetics, and evolutions. Genetic algorithms use techniques inspired by evolutionary biology. They represent an intelligent exploitation of a random search within a defined search space to solve a problem. It has been proven to be an effective tool to solve complex search problems with large solution spaces. In order to define a typical genetic algorithm, it is required a genetic representation of the solution domain and a fitness function to evaluate the solution domain. Genetic Algorithms start with an initial set of random solutions called population. Each individual in the population, called a chromosome, is an encoded string of symbols representing a solution, which may be feasible or infeasible.

Simon G. M. Koo et al. [3] proposed a genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks, which enhances the decision process performed at the tracker for transfer coordination. It increases content availability to the clients from their immediate neighbours. This neighbour selection strategy is being used for BitTorrent P2P network.

2.2. Based on Distributed Hash Tables (DHT) structures

Many systems locate nodes in the topology based on key-based graph structures such as CAN [8], Chord [9], Pastry [10] and Tapestry [11]. Their files are associated with a key (produced, for instance, by hashing the file name) and each node in the system is responsible for storing a certain range of keys. When a node joins the network, takes a key as input, and routes a message to the node responsible for that key, in response. Nodes have identifiers which are taken from the same space as the keys (i.e., same number of digits). Each node maintains a routing table consisting of a small subset of nodes in the system (their neighbours). In DHT structures, neighbours are selected as a function of the values of the key of the new node. When a node receives a query for a key for which it is not responsible, the node routes the query to the neighbour node that makes the most closest towards resolving the query.

The number of neighbours differs for each network. In Tapestry and Pastry a node has O(log n) neighbours, while in CAN has O(d) neighbours (d is the dimension of the toroidal space). Chord maintains two sets of neighbours. Each node has a successor list of k nodes that immediately follow it in the key space. There is a finger list of O(log n) nodes spaced exponentially around the key space.

These systems do not take care of the underlying network, so a neighbour of a node could be very far (in terms of Round Trip Time, RTT).

2.3. Based on the Internet underlying network

Trying to achieve the goal of having neighbours close “logically”, several systems have been proposed.

Plethora [12] is based on a two-level overlay architecture. The global overlay serves as the main data repository, and there are several local overlays that serve as caches to improve access time to data items. Local overlays contain nodes which IP are closer. Nodes that are in the same Autonomous System (AS) should be in the same local overlay together with nodes of neighbouring ASs. The global overlay is implemented using any prefix-based DHT system. It is used as the main repository and helps direct nodes to local overlays where they belong. The authors of this work adopt Pastry as Plethora’s underlying algorithm to support the local overlay. A node builds a list of ASs over the time that can be presented in its local overlay. Authors also propose to build this list using traceroute to some of the nodes in the node’s state tables. The node can measure the delay to each member of its routing table using the traceroute. If the delay is less than a system parameter, it includes the AS of the probed node in its neighborhood list. In each local overlay there is a node, the local overlay leader, which controls the number of nodes in the local overlay. Each node in a local overlay maintains a pointer to its current leader to determine if the leader has departed or failed.

T-DHT [13] is a scalable and distributed algorithm for the construction of distributed hash tables which are strongly oriented to the underlying network topology. The system is based on a virtual coordinate system. To build it, three (or more) reference nodes are randomly selected. Each node triangulates its position in the virtual coordinate space from these reference nodes. The virtual coordinate space is node’s position in the network topology, but not the physical position. Inspired by the Content Addressable Network (CAN), the authors of the work construct a two-dimensional DHT on the top of the virtual coordinate system. The coordinate space is divided among the participating nodes. Each node maintains a rectangular area around its position in the virtual coordinate system. It also maintains a routing table containing the path to its neighbours in the coordinate system and the area each neighbour maintains. A node may have links to nodes which are not direct neighbours in the hash table. Three steps must be performed to join the T-DHT. First, the node wishing to join must first find a node which is already in the T-DHT. Then, via T-DHT routing, it finds the node maintaining the zone of its position in the virtual coordinate system. This zone is equally split between the two nodes. Finally, the new member informs its neighbours about its presence.

To bootstrap, the first reference node maintains the whole DHT. When the virtual coordinates are assigned, it announces its T-DHT membership to its neighbours. Next, the neighbours join the distributed hash table and themselves announce their membership to their neighbours. The joining node only needs to contact the node maintaining the corresponding area.

2.4. Based on the RTT to the neighbour

It is important to achieve lower delays to exploit proximity in the underlying network in terms of RTT. Otherwise, each overlay hop has an expected delay equal to the average delay between a pair of random overlay nodes, which stretches route delay by a factor equal to the number of overlay hops and increases the stress in the underlying network links. Proximity neighbour selection (PNS) can be used to achieve low delay routes and low bandwidth usage. It selects routing state entries for each node from among the closest nodes in the underlying topology that satisfy constraints required for overlay routing. On the other hand, finding effective ways to produce proximity information is crucial for overlay networks to route efficiently. The proximity information can be used to partition nodes into clusters or to estimate distances among them.

M. Castro et al. presented in [14] a proximity neighbour selection (PNS) using heuristics approximations (called constrained gossiping (PNS-CG)). It can be used over DHT structures such as Pastry or Tapestry. The flexibility in the choice of nodeIds to fill routing table slots can be exploited to implement PNS effectively. Proximity neighbour selection picks the closest node in the underlying network from among those whose nodeIds have the required prefix. The proximity metric used in the definition of closest is RTT. In PNS-CG, when a new node x with nodeId X joins the overlay, it must contact an existing overlay node which routes a message using X as the key. The new node obtains the n^{th} row of its routing table from the node encountered along the path from the existing overlay node to X whose nodeId matches X in the first n-1 digits. Then, it updates other node's routing tables. When x's resulting routing table is updated, the closest node can be found using a specified algorithm designed by them.

It performs both low delay routes and low bandwidth usage with low overhead than other protocols such as Pastry and Tapestry, although the algorithm uses the routing state maintained by Pastry to locate nearby seed nodes for joining the network.

Others systems, such as the one presented by X. Zhichen in [15], locate new nodes in the topology using landmark clustering, as a preselection process to find nodes that are possibly close to a given node, and then perform RTT measurements to identify the actual closest node. Each node is assigned a landmark number that reflects its physical position in the network. Landmark clustering is based on the intuition that nodes close to each other are likely to have similar distances to a few selected landmark nodes. A node uses its landmark number as the DHT key to access relevant proximity information. To effectively use the proximity information generated, the information of the system is stored as soft-state in the system itself. To guide the placement of proximity information landmark clustering is used. Later, this information is used for nodes to discover other nodes physically near.

2.5. Based on their geographical location

The type of networks where it is more needed to chose the neighbours geographically because of their features are wireless ad-hoc and sensor networks. In these networks, connections are established only if they are closed, because of their coverage area limitation. In these types of networks, all nodes must know their own positions, either from a GPS device, if outdoors, or through other means and its neighbours' positions. There use to be a location registration and lookup service that maps node addresses to locations. If a node knows its neighbours' positions, the locally optimal choice of next hop is the neighbour geographically closest to the packet's destination. Forwarding in this regime follows successively closer geographic hops, until the destination is reached.

The self-describing nature of position is the key to geography's usefulness in routing. The position of a packet's destination and positions of the candidate next hops are sufficient to make correct forwarding decisions, without any other topological information. Examples given are the Routing with guaranteed delivery in ad hoc wireless networks presented by P. Bose et al. in [16] (also called GFG algorithm), they described a routing algorithm that guarantees delivery of messages in MANETs, and GPSR geographic routing algorithm presented by Karp, B. in [17] (they transformed GFG algorithm into a protocol). In geographic routing packets are stamped with the positions of their destinations. Their graph is planar, that is, there are enclosed polygonal regions bounded by edges. In planar graphs, there could be loops when the destination is disconnected.

Previous mechanisms do not consider peer grouping to structure the network and in none of them connections are established using nodes' capacity.

3. Group-Based Architecture

The proposed architecture is based on a one-level architecture. Each time an event occurs, the update is propagated through the group. Each group is composed by three types of sensors. The central sensor delimits the area of the group (it has the same function than other sensors in the group). The intermediate sensor that is the one responsible for routing the information received from more external sensors to the central sensor of the group (they allow a fast convergence when a change in the network takes place). Finally, the border sensor is the responsible for routing the information from inside its group to other groups or for receiving information from other groups and distributing it inside its own group (they are very important because they give the group's boundary).

A sensor sends information to its group using the Reverse Path Forwarding (RPF) algorithm. This algorithm is also used to send data to the border sensor to reach neighbouring groups. Each group has an RPF database but, when this information has to be sent to a specific group, it is directly routed to the nearest border sensor to this group. When the sensor in the neighbouring group receives the information, it routes it to all the sensors in its group. As the system is based on groups, the information is sent quickly to other groups (through the shortest path from the border sensor). More information about how it works and its application areas can be found in our paper in [18].

Figure 1 shows a 3D vision of the proposed architecture, but just in one group. Border sensors are in light gray, intermediate sensors are in dark gray and the black node is the central sensor.

Nowadays there are numerous types of wireless sensors. Their coverage varies so much. Consequently, we have studied the number of sensors needed to cover a specific area according to their coverage radius. The observed coverage radius in existing networks can be divided in 9 main groups as it is shown in table 1. Figure 2 shows the number of sensors needed for cover different areas according to the coverage radius of the used sensors. It is given for each group in table 1. Despite of the sensor's range, we have defined the area of a group by the diameter of the group, so there will be a maximum number of hops between the most remote sensors. The procedure maintenance is explained later.

3.1. Neighbour selection algorithm

Each sensor has 3 parameters (*sensorID*, *groupID*, λ) that characterize the sensor. Let λ parameter be the sensor capacity that depends on the sensor's upstream and downstream bandwidth (in Kbps), its number of available links (*Available_Con*) and its maximum number of links (*Max_Con*), its % of available load and its energy consumption. It is used to determine the best sensor to connect with. The higher the λ parameter, the best sensor to connect with is. It is defined by equation 2.

$$\lambda = \frac{\left(BW_{up} + BW_{down}\right) \cdot Available\_Con \cdot L + K_2}{Max\_Con} \cdot \sqrt{1 - \frac{E^2}{K_1}} \qquad (2)$$

Where $0 \le Available\_Con \le Max\_Con$. L is the available load and E is the energy consumption. L and E values vary from 0 to 100, according to the state of the sensor. An energy consumption of 0 indicates it is fully charged and when it has a value of 100, indicates it is fully discharged. K_1 defines the minimum value of energy remaining in a sensor to be suitable for being selected as a neighbour. K_2 gives λ values different from 0 in case of L=0 or *Available_Con*=0. The root is out of the division because when the sensor is fully discharged, λ parameter has to be 0. We have considered K_2=100 to get λ into desired values. Figure 3 shows λ parameter values when the maximum number of links for a sensor is 16, for a bandwidth value of 2 Mbps, as a function of its available number of links for different available energy values of the sensor. Sensor's load is fixed to 50%. It shows that higher Energy values give higher λ parameter, so the sensor is more likely to be chosen as a neighbour, in case of less available connections for the same energy, the higher with most available connections is chosen.

Figure 4 shows λ parameter values when the maximum number of links for a sensor is 16 and all have the same available number of links (*Available_con*=6) as a function of the sensor available energy for different bandwidth values. Sensor's load is fixed to 80%. It shows that as the Energy is being consumed, λ parameter is being lower, but when it gets the 80% of consumption, it decreases drastically, so the sensor is more likely to be chosen as a neighbour, in case of more available energy. Figure 4 also shows that a sensor with higher bandwidth is preferred.

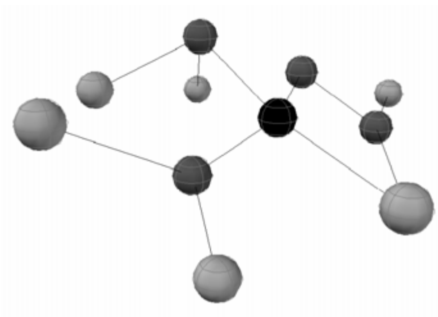


Figure 1. Proposed architecture topology example.

Group	Covarage radius, in meters
Group 1	1
Group 2	2
Group 3	5
Group 4	10
Group 5	20
Group 6	50
Group 7	100
Group 8	200
Group 9	500

Table 1. Groups depending on their coverage area.

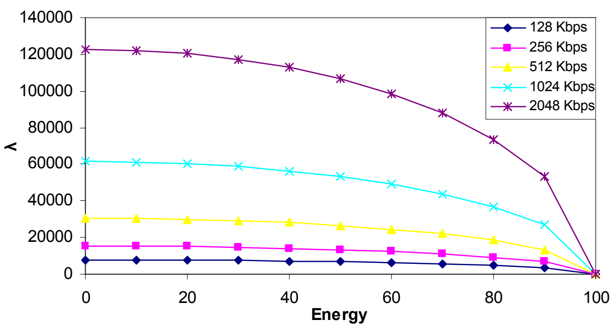


Figure 4. λ values as a function of the Energy of the sensor.

3.2. Protocol

When a new sensor joins the sensor network, it sends a discovery message (called *helloGroup* message) in order to join a group. If there is no response from any sensor for a pre-established time interval (in our case is 3 seconds), the sensor considers itself as a central sensor of a group in the network, and it will take the value *groupID*=1 and *sensorID*=1 (later, if existing sensor groups are joined, they will arrange a new *sensorID* for the group with less number of sensors). When the sensor receives *helloGroup ACK* messages from several candidate neighbours (they could be from different groups), it takes the RTT from each of them and puts a time stamp in their reply. Sensors which are in a distance of 15 hops to the central node will no reply. Then, the new sensor chooses the best sensor from the same group to have a

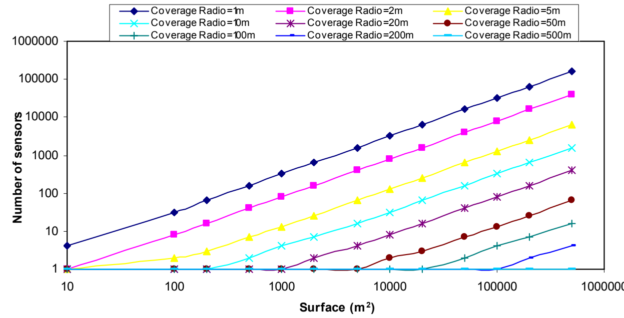


Figure 2. Number of sensor according to the coverage radius and the area to cover.

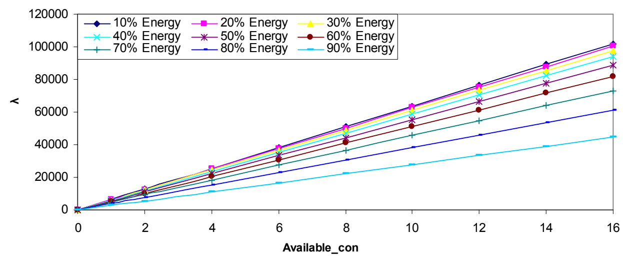


Figure 3. λ parameter values as a function of the available connections of the sensor.

link with (this election is taken according to the λ parameter which is included in the *helloGroup ACK* message and to the RTT). As the *groupID* is in the *helloGroup ACK* message, the new sensor will know which group it has joined (the one with lowest RTT) and it will store this value. If the sensor which sent the *helloGroup ACK* message doesn't receive a reply (*okGroup* message) in a limited period of time, it means that it has not been elected as a neighbour. Finally, new neighbours will reply with the *okGroup ACK* message with the assigned *sensorID* and indicating the link has been established. Sensors will send *keepalive* messages periodically to their neighbours. If a sensor does not receive a *keepalive* message from a neighbour for a dead time, it will remove this entry from its database and will start the group update process. All the process described is shown in figure 5. New sensors and failures in the sensor network have to be known by all the nodes, so, between all the elected neighbours there will be one, called responsible, that will send the information about the new node, using *newSensor* messages, to the central node. Then, central sensor could be changed because the central reference of the group has been moved (this decision is taken using the value of the diameter of the group). When it occurs, the sensors in the group will be advised by a *changeCentral* message. To provide fault tolerance for the central sensor, it calculates which is the best candidate and sends it *keepaliveCentral* messages periodically. In case of changes, updates are distributed using RPF algorithm.

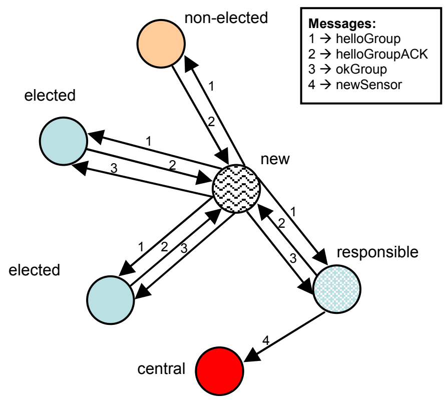


Figure 5. Message Exchange when a node joins the group.

4. Comparative and conclusions

Neighbour selection has been a hot topic for several years in many research topics, but there has not been any research where the nodes, which have to select its neighbours, are placed in different groups. We have shown a state of the art of the main existing neighbour selection algorithms. Some are based on search techniques, there are others based on DHTs, others are based on GPS and, finally, several are based on RTT which give fastest replies. Our proposal is based on RTT and on a λ parameter which combines several parameters such as bandwidth, load, energy, available number of connections and on the maximum number of connections of the sensor. It gives us fastest replies while introduces major parameters when the neighbour selection decision takes place. The number of node's neighbours could be limited only by establishing a maximum in all previous works, whereas in our proposal as the number of connections of a node gets higher, it is less probable to be chosen as a neighbour. In our system, λ parameter allows to distribute the load of the network between groups while it balances the number of sensors in the groups, distributing them uniformly in all the groups of the network.

5. References

[1] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, W. Willinger, "Network Topology Generators: Degree-Based vs. Structural", Proceedings of the ACM SIGCOMM, August 2002.

[2] L. Zou, E. Zegura, M.H. Ammar, The effect of peer selection and buffering strategies on the performance of peer-to-peer file sharing systems. Proceedings of 10th Int'l Symposium on Modeling, Analysis and Simulation of Computer and Telecom. Systems, Fort Worth, TX, 2002.

[3] S.G.M. Koo, K. Kannan, C.S.G. Lee, A genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks. Proceedings of the 13th IEEE International Conference on Computer Communications and Networks, ICCCN'04, Chicago, IL, October 2004, pp. 469–474.

[4] Simon G.M. Koo, Karthik Kannan and C.S. George Lee, On neighbor-selection strategy in hybrid peer-to-peer networks, Future Generation Computer Systems Volume 22, Issue 7, August 2006, Pages 732-741.

[5] D.S. Bernstein, Z. Feng, B.N. Levine, S. Zilberstein, Adaptive peer selection. Proc. of the 2nd Int'l Workshop on Peer-to-Peer Systems, Berkeley, CA, Feb. 2003.

[6] J. Byers, J. Considine, M. Mitzenmacher, S. Rost, Informed content delivery across adaptive overlay networks. ACM SIGCOMM 2002, Pittsburgh, PA, August 2002.

[7] S.G.M. Koo, C.S.G. Lee, K. Kannan, A resource-trading mechanism for efficient distribution of large-volume contents on peer-to-peer networks. Proceedings of the 14th IEEE International Conference on Computer Communications and Networks, pp. 428–433. San Diego, CA, October 2005.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A Scalable Content-Adressable Network, ACM Sigcomm 2001

[9] I. Stoica, R. Morris, D.Karger, F.Kaashoek, H. Balakrishnan, Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications, ACM Sigcomm 2001

[10] A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), heidelberg, Germany, pages 329-350, Noviembre, 2001

[11] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, J.D. Kubiatowicz. Tapestry: a resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications, Vol. 22, Issue 1. Pp 41- 53, 2004

[12] R. A. Ferreira, S. Jagannathan, A. Grama. Locality in structured peer-to-peer networks, Journal of Parallel and Distributed Computing, Vol. 66, Issue 2. Pp. 257-273. Feb. 2006.

[13] Olaf Landsiedel, Katharina Anna Lehmann, Klaus Wehrle, T-DHT: Topology-Based Distributed Hash Tables. Fifth IEEE International Conference on Peer-to-Peer Computing. Pp. 143-144. 2005.

[14] M. Castro, P. Druschel, Y. C. Hu and A. Rowstron, Proximity neighbor selection in tree-based structured peer-to-peer overlays, Technical Report MSR-TR-2003-52, Microsoft Research, Microsoft Corporation. 2003.

[15] Xu, Z., Tang, C., Zhang, Z. Building topology-aware overlays using global soft-state. Proceedings of the 23rd Int'l Conference on Distributed Computing Systems. May 2003.

[16] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. Proceedings of ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, USA, August 1999.

[17] Karp, B., Kung, H.T. GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking. Pp. 243–254. New York, USA, 2000,

[18] J. Lloret, M. Garcia and J. Tomas, A Group-Based Architecture for Wireless Sensor Networks, Int'l Conf. on Networking and Services, Athens (Greece), June 2007.

A Group-Based Protocol for Large Wireless AD-HOC and Sensor Networks

Jaime Lloret[1], Miguel Garcia[2], Fernando Boronat[3] and Jesus Tomás[4]

Department of Communications, Polytechnic University of Valencia
Camino Vera s/n, 46022, Valencia (Spain)
{[1]jlloret, [3]fboronat, [4]jtomas}@dcom.upv.es, [2]migarpi@posgrado.upv.es

***Abstract*— Many routing protocols for ad-hoc networks and sensor networks have been designed, but none of them is based on groups. It is known that grouping nodes gives better performance to the group and to the whole system, thereby avoiding unnecessary message forwarding and additional overheads. We propose an approach where the network is split into several groups of sensors where connections between groups are established as a function of the proximity and the neighbor's available capacity (based on the sensor's energy). In this paper the network architecture is described with its mathematical description and the messages that are needed to proper operation. It is also simulated how much time is needed to propagate information between groups. A comparison with another group-based architectures is shown. The application areas for our proposal could be rural and agricultural environments in order to detect plagues or fires and to propagate it to neighboring areas, or for military purposes to propagate information between neighboring squads.**

***Index Terms*— Group-based protocol, Group-based architecture, Group-based routing algorithm, WSN.**

I. INTRODUCTION

THE physical network topology defines how the nodes on a network are physically connected, i.e., the physical layout of the devices on the network. We can distinguish three types of network topologies:

1. Centralized Networks. They are topologies in which there is no direct connection between nodes and all nodes' messages are mediated by a central node. Centralized topologies have been used for many types of networks [1].

2. Decentralized Networks. All nodes have the same responsibility and functionality in the network. Every node is able to connect directly with all other nodes, and it can be a server, a client, and a router. Many types of networks have a decentralized topology such as pure P2P networks, ad-hoc and sensor networks, grids and others.

3. Partially Centralized Networks. Some nodes have higher roles which form the backbone of the network (the higher logical layer) and they are needed to run the system. Nodes with lower role are called leaf nodes and will be placed in the lower logical layer. There is a kind of hierarchy where higher layer nodes organize, control or gather data from lower layer nodes. They have been used for different types of networks such as satellite networks [2] and even for WSNs [3].

Many routing protocols can be applied to sensor networks. They can be classified into two groups [4] [5]. The first group is formed by protocols based on the network topology and it could be broken down into three subgroups such as plane routing (all sensors in the network have the same role and perform the same tasks), hierarchical routing (it has been designed for energy saving purposes) and position-based routing (all data is routed through the sensors depending on their position). The second group does not have into account the structure of the network. It can be broken down into five subgroups such as Multipath Routing Protocols, Query-Based Routing Protocols, Negotiation-Based Routing Protocols, QoS protocols and data coherent/incoherent processing based protocols. But none of the routing protocols aforementioned are group-based. We propose to divide the sensor network into several groups and if a sensor has to send data to other groups, when this data arrives to one sensor from a group it is propagated to the rest of sensors in its group.

The rest of the paper is structured as follows. Section 2 describes group based architectures. The description of our proposal, its analytical model and the formulas used to select sensor's neighbors are shown in section 3. Joining sensors and failures, protocol operation and its messages are shown in section 4. All simulations we have done for our protocol are shown in section 5. In section 6, we have compared our proposal with another existing planar group-based architecture. Finally, section 7 summarizes the results and gives our future research.

II. GROUP-BASED ARCHITECTURES

First, we have to distinguish between a groupware architecture, where all nodes collaborate towards the correct operation and the success of the purpose network, and group-based architecture, where the whole network is broken down into groups and each group could perform different operations. In a wireless group-based architecture, a group consists of a set of nodes that are close to each other (in terms of geographical location or in terms of round trip time). These groups could have a link if a node from a group is near a node from another group. The distance between two nodes is given by the network latency or the round trip time. The main goal in a wireless group-based topology is the network protocol and the group management, that is the design of an efficient algorithm for a new node to find its nearest (or the best) group to join in. Then, some important issues must be designed: (i) How to build neighboring groups, and (ii) a protocol to exchange messages between neighboring groups. The performance of the group-based network highly depends on

978-1-4244-2067-4/08/$25.00 ©2008 IEEE.

the efficiency of this nearby group locating process. All found works, where nodes are divided into groups and connections are established between nodes from different groups, have been developed to solve specific issues such as distribution service in multimedia networks [6], group-based games over NAT/Firewalls [7], and so on.

We can distinguish two types of group-based topologies: Planar group-based topologies and layered group-based topologies. In planar group-based topologies, all nodes perform the same roles and there is just one layer. However, in some works there is a directory server or a rendezvous point (RP) for content distribution coordination. Nodes from layered group-based topologies could have several roles (2 roles at least). Depending on which type of role they are running, they will become to a specific layer. All nodes in the same layer will have the same role. There will be connections between nodes from the same layer and from adjacent layers.

There are several differences between both group-based topologies. While layered group-based topologies grow structured organized by upper layers, planar group-based topologies grow unstructured without any organization. In layered group-based topologies anyone can know exactly where each group is and how to reach it. Otherwise planar group-based topologies, every time a node wants to reach other group, the message should travel through many unknown groups due to groups join the network as they appear. Delays between groups in layered group-based topologies could be lower because connections between groups can be established taking into account this parameter, otherwise, in planar group-based topologies connections between groups are established by group's position, their geographical situation or because of their appearance in the network. Layered networks address several complexities because nodes could have several types of roles and fault tolerance for every layer must be designed. On the other hand, planar networks are more simplex because all nodes have the same role. In order to have scalability, layered group-based topologies must add more layers to its logical topology, while planar group-based topologies could grow without any limitation, just the number of hops of the message.

Cluster-based networks are a subset of the group-based networks, because each cluster could be considered as a virtual group. But a group-based network is capable of having any type of topology inside any group, not only clusters. In cluster based architectures each cluster has adjacencies with other clusters and all clusters have the same rules. A cluster can be made up of a Cluster Head node, Cluster Gateways and Cluster Members [8] [9]. The Cluster Head node is the parent node of the cluster, which manages and checks the status of the links in the cluster, and routes the information to the right clusters. The rest of the nodes in a cluster are cluster members. In this kind of network, the Cluster Head nodes have a total control over the cluster and the size of the cluster is usually about 1 or 2 hops from the Cluster Head node. A cluster member does not have inter-cluster links.

We can also find in the literature a routing protocol based on zones. It is the Zone Routing Protocol (ZRP) [10] [11]. Each node proactively maintains routing information for a local neighborhood (routing zone), while reactively acquiring routes to destinations beyond the routing zone. ZRP and our proposal have several common features, e.g. they could be applied over any type of routing protocol, they scale well and the information is sent to border nodes in order to reach destinations outside their zones. The main difference between them is that in ZRP each node maintains a zone and the nodes in that zone have different nodes in their zone while in our proposal all the nodes that form a group have the same nodes in their group.

On the other hand, we will not consider other works of groups systems such as the following. The community based mobility model for ad hoc network research presented in [12], because although the network is organized in groups, and nodes can move from one host to another, there is not any connection between border nodes from different groups. The landmark hierarchy presented in [13] because although there is a node with higher role which has connections with nodes from other groups, its leaf nodes do not have. Another example similar to the last one is the BGP routing protocol architecture [14]. Finally, we will not consider moving groups such as Landmark Routing Protocol (LANMAR [15]), where the set of nodes move as a group, so the group can enlarge or diminish with the motion of the members.

III. ARCHITECTURE DESCRIPTION

A. Architecture Operation

We propose a structure of sensors based on the creation of sensor groups with the same functionality in the network. For every group exists a central sensor that limits the zone where the sensor from the same group will be placed, but its functionality is the same that the rest of the sensors. Every sensor has a *sensorID* that is unique in its group. The first sensor in the network acquires a group identifier (*groupID*) that is given manually, using GPS, using a wireless location system or through other means. New joining sensors will know their group identifier from their new neighbors. Border sensors are, physically, the edge sensors of the group. When there is an event in one sensor, this event is sent to all the sensors in its group in order to take the appropriate actions. All sensors in a group know all information about their group. Border sensors have connections with other border sensors from neighbor groups and are used to send information to other groups or to receive information from other groups and distribute it inside. Because it is needed a fast routing protocol, we have chosen SPF (Shortest Path First) routing algorithm [16] to route information, but it can be changed by other routing protocol depending on the network's characteristics. When the information is for a sensor of the same group it is routed using the *sensorID*. Every sensor runs SPF algorithm locally and selects the best path to a destination based on a metric.

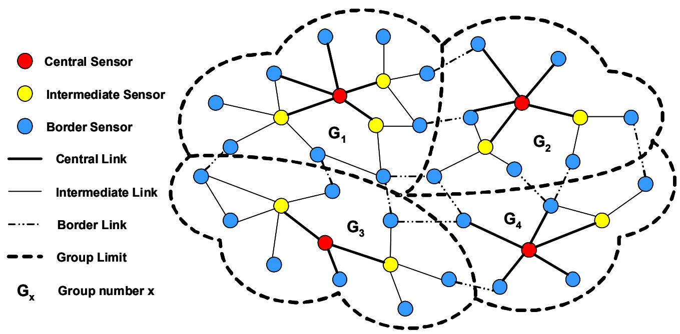


Fig. 1. Proposed architecture topology.

TABLE I
EQUATIONS OF THE PARAMETERS

Parameter	Equation
λ	$\frac{Available\_Con \cdot L + K_2}{Max\_Con} \cdot \sqrt{1-\frac{E^2}{K_1}}$
C	$\frac{T \cdot K_3}{\lambda}$
$Metric$	$\sum_{i=1}^{r} C_i$
n	$\sum_{i=1}^{k} \lvert V_k \rvert$

When the information has to be sent to other groups, the information is routed directly to the closest border sensor to the destination group using the *groupID*. When a sensor from destination group receives the information, it routes it to all sensors in its group using Reverse Path Forwarding Algorithm [17]. Links between border sensors from different groups are established primarily as a function of their position, but in case of multiple possibilities, neighbors are selected as a function of their capacity. In order to establish the boundaries of the group, we can consider two choices: (i) limiting the diameter of the group to a maximum number of hops (e.g., 30 hops, as the maximum number of hops for a tracer of a route), and (ii) establishing the boundaries of the area that it is wanted to be covered. Figure 1 shows the proposed architecture topology.

B. Analytical Model and Neighbor Selection

Every sensor has 3 parameters (sensorID, groupID and λ) that characterize the sensor. Let λ parameter be the sensor capacity that depends on the sensor's number of available links (*Available_Con*) and its maximum number of links (*Max_Con*), its % of available load and its energy consumption. It is used to determine the best sensor to connect with. The higher the λ parameter, the best sensor to connect with is. λ equation is shown in table I, where L is the available load and E is the energy consumption. Their values vary from 0 to 100. E=0 indicates it is fully charged, so λ parameter is 0 and E=100 indicates it is fully discharged. K_1 defines the minimum value of energy remaining in a sensor to be suitable for being selected as a neighbor. K_2 gives different λ values from 0 in case of L=0 or *Available_Con*=0. We have considered K_2=100 to get λ into desired values. Figure 2 shows λ parameter values when the maximum number of links for a sensor is 16 as a function of its available number of links for different available energy values of the sensor. Sensor's load is fixed to 50%.

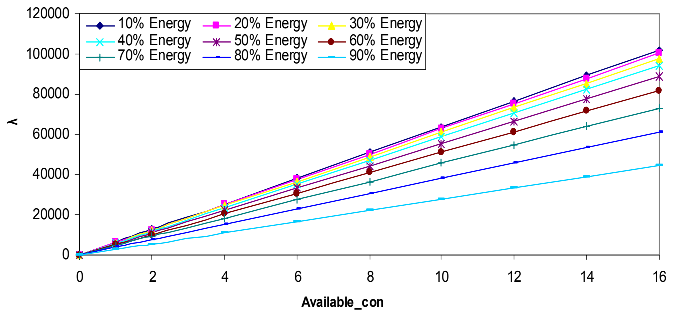


Fig. 2. λ parameter values with number of links variation.

We have defined the cost of the i^{th}-Sensor as the inverse of the i^{th}-Sensor λ parameter multiplied by T (the delay of its reply in msec). It is shown in table I. $K_3=10^3$ gives $C\geq 1$. The metric for each route is based on the hops to a destination (r) and on the cost of the sensors (C_i) in the route as it is shown in table I. The metric gives the best path to reach a sensor.

Let G = (V, λ, F) be a network of sensors, where V is the set of sensors, λ is the set of their capacities ($\lambda(i)$ is the capacity of the i-th sensor and $\lambda(i)\neq 0\ \forall$ i-th sensor) and F is the set of links between sensors. Let k be a finite number of disjoint subsets of V, so $V=\cup V_k$, and there is not any sensor in two or more subsets ($\cap V_k=0$), and let be n=|V| (the number of sensors in V), the equation given for n is shown in table I.

Every V_k has a central sensor, several intermediate sensors and several border sensors as is shown in equation 1.

$$n = 1 + n_{intermediate} + n_{border} \quad (1)$$

Now we can describe the whole network as the sum of all these sensors from all groups as it is shown in equation 2.

$$n=\sum_{i=1}^{k}\lvert (n_{central}+n_{intermediate}+n_{border})_k \rvert = k+\sum_{i=1}^{k}(\lvert n_{intermediate}\rvert)_k+\sum_{i=1}^{k}(\lvert n_{border}\rvert)_k \quad (2)$$

On the other hand, the number of links in the whole network m=|F| depends on the number of groups (k), on the number of links in each group (k_m) and on the number links between border sensors. Equation 3 gives m value for a physical topology.

$$m=\sum_{i=1}^{k}\left(k_l+\frac{1}{2}k_b\right) \quad (3)$$

Where k_l is the number of links inside the group k and k_b is the number of external links of the group k.

IV. PROTOCOL OPERATION AND MESSAGES

This section describes the designed messages and how the designed protocol operates.

A. Group Creation and Maintenance

Let a new sensor be in the network (it could be the first). It sends a hello message (called *helloGroup*) in order to join a group. If there is no response from any sensor for 3 seconds, the sensor considers itself as a central sensor of a group in the network, and it will take the value *groupID*=1 and *sensorID*=1. When the sensor receives *helloGroup ACK* messages from several candidate neighbors, first it puts a time stamp in their reply and chooses the best sensors to have a link with (this election is taken based on the λ parameter which comes in the *helloGroup ACK* message). The time stamp will be used to calculate *C* parameter. Responses received after 3 seconds will be discarded. In case of receiving replies from sensors of different groups, it will choose the group which replies have the highest average λ parameter, so it will take into account replies only for that group. Then, the sensor will send an *okGroup* message to the selected neighbors, and the neighbors will reply with the *okGroup ACK* message with the assigned *sensorID* and indicating the link has been established. Sensors will send *keepalive* messages periodically to their neighbors. If a sensor does not receive a *keepalive* message from a neighbor before the dead time, it will remove this entry from its database and will start the group update process. As the *groupID* is in the *helloGroup ACK* message, the new sensor will know which group has joined. Finally, the neighbor sensor will send a *newSensor* message to the central sensor, to run the algorithm for changing the central sensor if it is needed.

Links between border sensors from different groups are established as a function of their replying delay and the λ parameter of the replying sensors, but it could be changed by an algorithm using sensor's position or choosing the neighbor with the lower distance (in number of hops) to the central sensor. If we base our proposal on the λ parameter, we will distribute the load of the network between groups, but if we base our proposal on sensor's position or choosing the neighbor with the lower distance to the central sensor we will balance the number of sensors in the groups.

When a new sensor joins the group, the central sensor of the group could be changed. The procedure designed for changing the central sensor is as follows. We define the group diameter (d_{group}) as the shortest number of hops, between the two most remote sensors in the group (in our case, $d_{group} \leq 30$).

When there is a change of the central sensor of a group, all the sensors in the group must be advised. In order to update all sensors in the group, the new central sensor will send a *changeCentral* message to indicate the new central sensor and the distance from it to the sensor processing this control packet. This update is distributed using the RPF algorithm. Once the links between neighbors are established, every sensor sends *keepalive* messages periodically to its neighbors. Figures 3 and 4 show the procedure when the central sensor changes and when it doesn't.

We have proposed two choices to establish the boundaries of the group:

1) When the boundaries of the group are the same of the area that it is wanted to be covered, border sensors are known using GPS.
2) When the boundary of the group is limited by the diameter of the group, the maximum number of hops from the central sensor must be known. Every time a new sensor joins a group, it receives the newSensor ACK message with the number of hops to the central sensor. When it achieves the maximum number of hops, the sensor is marked as a border sensor, and it will inform new joining sensor that they must create a new group.

B. *Leavings and Fault Tolerance*

When a sensor leaves the the group, it will send *sensorDisconnect* message to its neighbor sensors. They must reply with a *sensorDisconnect ACK* message and send to the central sensor the *sensorDisconnect* message. The central sensor distributes the update information using RPF algorithm. If the neighbor sensor doesn't have links with other neighbors, it must start a new connection process sending a *helloGroup* message. If the leaving sensor is the central sensor, it assigns the central sensor role to the best candidate to be the central sensor (in case of draw, it will choose the older one in the group), then it sends a *changeCentral* message to the group to inform them and leaves the group. When a sensor fails down, its neighbor sensors will know the failure because of the missing of its *keepalive* messages. The procedure is the same as when the sensor leaves the network voluntarily. The central sensor calculates which the best candidate is, and the neighbor sensor will be informed by periodical *keepaliveCentral* messages. New central sensor will distribute the update information.

V. SIMULATIONS

Let T_i be as the time needed by two sensors to communicate each other, and RTT as the mean value of the round trip time between both sensors. So, T_i can be calculated using the expression given in table II. The time needed to communicate a source sensor with a destination sensor in a different group is calculated using the expression given for $T_{max\_intergroup}$ in table II. Where *n* is the number of intermediate groups, t_{source_border} is the time needed to arrive from the source sensor to the border sensor in the same group, $t_{max\_intragroup\_i}$ is the time required to go through the i-th group, and $t_{border\_i\text{-}border\_i+1}$ is the time needed to transmit the information from the border sensor of a group to the border sensor of another group connected to the previous one.

We define t_p as the average propagation time for all the message transmissions between two sensors in the architecture. Its expression is shown in table II. Where *m* represents the number of sensors involved in the path minus one. Taking into account t_p, the time needed to transmit information from the source sensor to the border sensor of the same group (T_{source_border}) is defined as it is shown in table II. Where d_{source_border} are the number of hops needed to arrive form the source sensor to the border sensor of the same group. The maximum time to cross a group ($T_{max\_intragroup\_i}$) is defined by the expression shown in table II. *i* indicates the group and the d_i is the number of hops in the group. On the other hand, the number of hops for *j* groups is shown in table II.

Replacing equations in Table II, we obtain equation 4.

$$T_{\max\_intergroup} = \left(d_{source\_border} + \sum_{i=1}^{n} d_i + n + 1 \right) \cdot t_p \qquad (4)$$

A. Time variation as a function of the number of hops to the border sensor when all the groups have the same number of hops

This simulation is done fixing the number of intermediate groups and the number of hops between source sensor and the border sensor of its group is varied. Then, we can observe what happens when the number of hops of the intermediate groups increases.

We have chosen the number of intermediate groups as 4. Considering that all the intermediate groups have the same number of hops, it means $d_1=d_2=d_3=d_4=d$, and introducing these values in equation 4 we obtain the expression 5.

$$T_{\max\_intergroup} = \left(d_{source\_border} + 4 \cdot d + 5 \right) t_p \qquad (5)$$

When we give higher values to d_{source_border} for each value of d, the maximum inter group time ($T_{max\_intergroup}$) rises lineally.

B. Time variation when the number of hops to cross the groups varies

This section studies what happens when we maintain the distance between source sensor and the border sensor of the source group constant and we vary the number of hops of the intermediate groups and for different number of groups. We fix the parameter d_{source_border} to a value of 10. Using equation 4, equation 6 is obtained.

$$T_{\max\_intergroup} = \left(11 + \sum_{i=1}^{n} d_i + n \right) \cdot t_p \qquad (6)$$

Now, we can vary d_i to observe the time needed to achieve the destination. Results are shown in figure 5. We can deduce that the number of groups in a network doesn't affect the connection time to a large extent when the mean number of hops to go through the groups is low. Nevertheless, when the mean diameter of the groups is big, when we increase the number of intermediate groups, the connection time rises too much. So, we can state that the mean diameter of the groups becomes more relevant in the calculation of the final connection time ($T_{max\_intergroup}$) for bigger networks.

In figure 6, we can observe how the connection time varies according to the number of groups for different number of hops. We have chosen d_{source_border} = 20, and we have varied the number of groups that will be crossed for different mean diameters of the groups, instead of varying the mean diameter of the groups.

C. Variation of the time for different number of groups and different distances between source and border sensors in the same group

In this section we analyze how the maximum inter group time varies when we maintain the mean diameter of the group as a constant value and vary the number of groups for different distances between source and border sensors of the same group. To perform this experiment, we have chosen 20 as the mean diameter of the groups. Equation 7 shows that the connection time depends on the distance between the source and border sensors in the same group and on the amount of groups in the network.

$$T_{\max\_intergroup} = \left(d_{source\_border} + 21 \cdot n + 1 \right) t_p \qquad (7)$$

Figure 7 shows the behavior of the $T_{max\_intergroup}$ as a function of n for several d_{source_border} values. On the other hand, we can observe in figure 8 that the maximum inter group time (t_p) increases when the number of intermediate groups increases. It is happening in all the analyzed cases. Nevertheless, as we can see, there is not a big difference in the final time when we have a large or short distance between source and border sensor (d_{source_border}). It means, it is more relevant the number of groups than d_{source_border} for having better $T_{max\_intergroup}$. This is an important subject to take into account when designing sensor networks.

D. Number of sensors as a function of the topology of the group

In order to limit the coverage of the group, let's suppose there are a maximum number of hops between the central sensor and the most distant border sensor, using the shortest path. In this option, the geographic coverage of the sensors in a group will depend on the amount of sensors in some specific part of the group topology and on its position. We define the influence area as all sensors near the central sensor, with an access time lower than T_{max}. We suppose the sensors are in a distance of d_{ij}. We can observe it in the figure 8, where T_{max} is specified according to the network and the distance d_{ij}, in number of hops. It will be the minimum distance between both sensors.

First, we have taken a maximum diameter of 30 hops. This indicates that there will have a maximum of 14 sensors between the central sensor and any border sensor. The maximum distance can vary depending on if we want to make the group larger or smaller, but it will depend on the needs of the project in which the protocol is applied. When an event is noticed by a sensor of the group, it will be forwarded immediately to the entire group. So, we have to look for a commitment between the number of sensors in the group and the number of groups needed to cover the entire sensor network, in order to minimize the convergence time. The diameter depends on the topology of the sensors in the group.

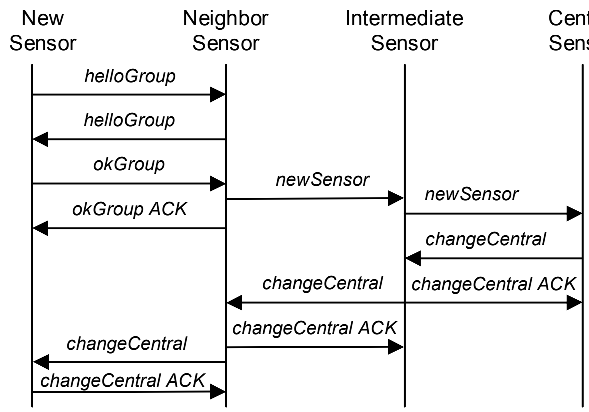


Fig. 3. Messages when central sensor changes.

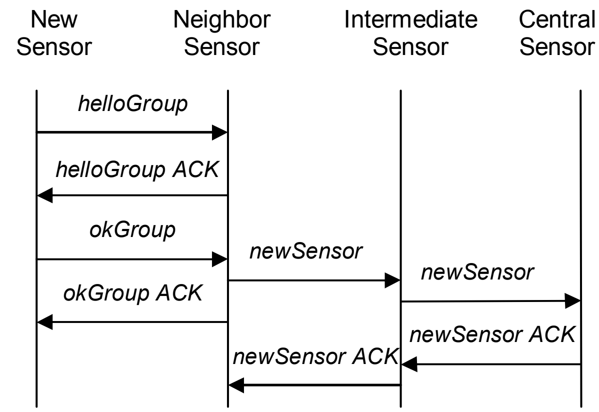


Fig. 4. Messages when it doesn't change.

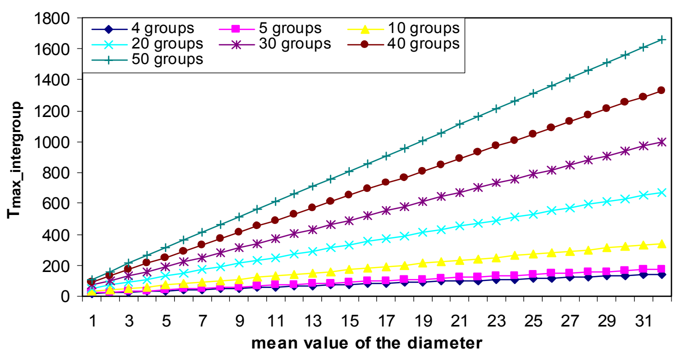


Fig. 5. $T_{max\_intergroup}$ variation according to the mean diameter of the groups.

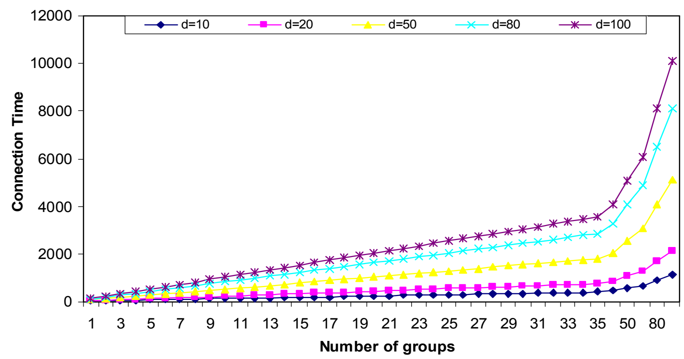


Fig. 6. Variation of the connection time according to the number of groups.

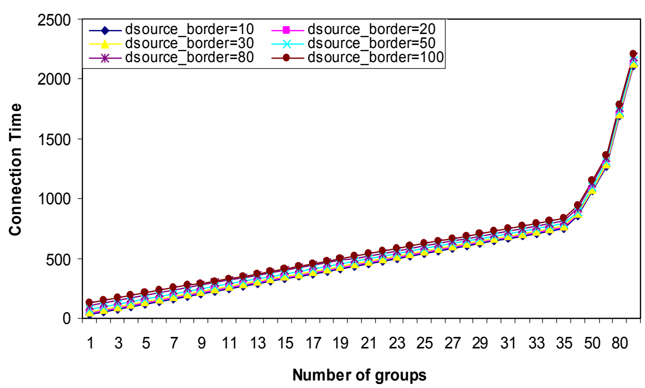


Fig. 7. Variation of the connection time according to the number of groups.

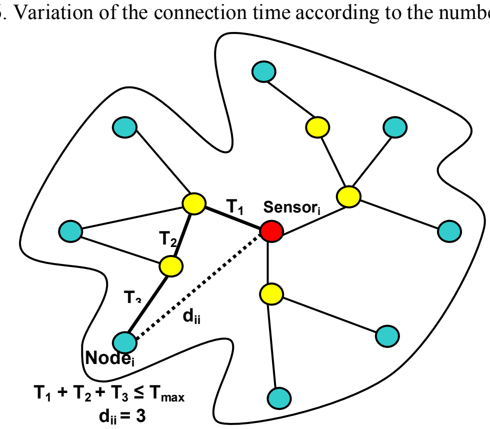


Fig. 8. Central Sensor influence area.

TABLE II
EQUATIONS FOR PARAMETERS BETWEEN DIFFERENT NODES

Parameter	Equation
T_i	$\frac{RTT_i}{2}$
$T_{max\_intergroup}$	$t_{source\_border} + \sum_{i=1}^{n} t_{\max\_intragroup\_i} + \sum_{i=1}^{n+1} t_{border\_i-border\_i+1}$
t_p	$\frac{\sum_{i=1}^{m} T_i}{m}$
T_{source_border}	$d_{source\_border} \cdot t_p$
$T_{max\_intragroup\_i}$	$d_i \cdot t_p$
d_i	$\sum_{j=1}^{d_j} d_j$

In a tree topology, the number of links is n-1, where n is the number of sensors in the tree. The diameter of the tree topology is shown in table III. *M* is the number of son nodes of a node in a tree topology (2 if binary, 3 ternary, etc.). The diameters of a 2D and of a 3D grid topology, with *n* nodes, are shown in table III. If we suppose that the network scales freely, without control, following the Zipf law, when we have to chose the maximum distance (d_{max}) we can base on the cut-off effect shown by R. Cohen et al. in [18] [19]. In networks where routing algorithms cross the network in few hops (such as Internet), *d* is defined as it is in table III. On the other hand, the logarithmic law states that the mean distance between two border nodes is equivalent to the diameter according to the law described in [20]. It can be calculated as it is in table III. Finally, there is a law that follows P2P partially decentralized networks. If we consider a TTL of 6 according to and considering a mean of 16 superpeer neighbors in every hop, there will be 96 border nodes for every central node. So, the diameter of the network can be calculated as it is shown in table III. Figure 9 shows the comparative between the studied topologies.

The lower diameter is Zipf law with a R coefficient of -2,45. Nevertheless, when there are less than 14 nodes in the network, the worst case isthe binary tree topology. Between 15 and 22 nodes, both binary tree and 2D grid topologies are the worst ones, and when there are more than 23 nodes, the worst case is the 2D grid topology.

VI. Architecture Comparison

In this section we are going to compare our proposal with a planar group-based architecture and with cluster-based networks.

There are very few works about planar group-based topologies. Xiang et al. proposed a locality-aware overlay network based on groups [21], and later, they presented a Peer-to-Peer Based Multimedia Distribution Service [6] based on this proposal. In their proposal, nearby nodes in the underlying network self-organize into application groups. Each peer contributes its local storage and I/O capacity to support multimedia distribution service to other peers. The system uses a locating method to join a nearest group or form its own group according to the group criterion. The nearby group for a new joining node is found based on a distance measurement using a global server cache, called rendezvous point (RP), and some boot nodes. Each group maintains a local node cache, which consists of nodes in the same group responsible for communications with nodes in other groups. The group also maintains information about its neighbor groups. The first host, called the leader, is responsible for updating information. The second host in the host cache will stand up and take over the leader's responsibilities in case of the leader failure. When a node in this architecture wants to communicate with a node from other group, the information is routed through the groups until it arrives to the destination.

As we have introduced in section 2, the main feature of cluster-based networks is that they use to have a cluster head that have connections with cluster heads of other clusters, giving a hierarchical architecture, and that all clusters have the same characteristics.

TABLE III
Diameters of some topologies that can be used in the group

Topology	Diameter
Tree	$d = 2\cdot\log_M\left[n\cdot(M-1)+1\right]-2$
2D Grid	$d = 2\cdot\left(\sqrt{n}-1\right)$
3D Grid	$d = 3\cdot\left(\sqrt[3]{n}-1\right)$
Zipf Law	$d \approx \log(\log(n))$
Logarithmic Law	$l_n \approx d \approx \ln\left(\dfrac{n}{2}\right)$
P2P partially decentralized	$d = \dfrac{n}{16}$

Table IV shows the comparison. Our proposal stands out because of its higher efficiency in the neighbor selection system (we have added the capacity parameter), lower management cost, high fault tolerance and very high scalability.

VII. Conclusions

A group-based architecture provides some benefits for the whole network such as the content availability is increased because it could be replicated to other groups, it provides fault tolerance because other groups could carry out tasks from a failed group and it is very scalable because a new group could be added to the system easily. On the other hand, a group-based network can significantly decrease the communication cost between end-hosts by ensuring that a message reaches its destination with small overhead and highly efficient forwarding. Grouping sensors increases the productivity and the performance of the network with low overhead and low extra network traffic.

In this paper we have described a group-based architecture proposal where links between groups could be established by physical proximity plus the neighbor sensor capacity. Its operation, maintenance and fault tolerance have been detailed. Messages designed to work properly have been shown. All simulations show its viability and how it could be designed to improve its performance. Finally we have compared it with another group-based logical architecture to show their differences.

The architecture proposed could be used for specific cases or environments such when it is wanted to setup a network where groups appear and join the network or by networks that are wanted to be split into smaller zones to support a large number of sensors. There are many application areas for this proposal such as rural and agricultural environments or even for military purposes. Now, we are programming the protocol for a specific wireless sensor device to test it over a real environment.

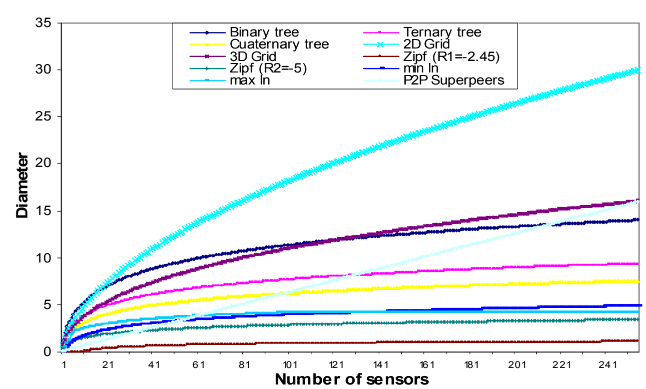


Fig. 9. Diameter according to the total number of nodes.

TABLE IV
GROUP-BASED TOPOLOGIES COMPARISON

	Locality-aware overlay netk (Z. Xiang et al.)	Our group-based architecture	Cluster-based architecture
Need of a Rendezvous Point	Yes	No	No
Sensors with higher role	No	Yes	No
Type of topology	Logical, but it could be implemented in physical	Physical, but is could be implemented in logical	Physical, but it could be implemented in logical
Neigbour selection	Proximity in the underlying network (IP)	Physical proximity + capacity	Physical proximity + capacity
Which group to join in	Based on rendezvous point decision + boot nodes	Based on neighbour discovery (time to reply or closest)	Based on neighbor discovery (time to reply or closest)
Leader	Yes	Yes	Low
Convergence time	Very little	Very much	Very much
Management cost	Medium because of the rendezvous point	Low	Very much
Fault tolerance	Very low (RP or boot nodes failure)	Very much	Very high (when a sensor finds a neighbor it joins the network)
Scalability	Very much (depending on the RP)	Very much	Physical, but it could be implemented in logical
Performance of the system	High	High	Physical proximity + capacity
Availability	Low (when boot nodes from a group are not available the group is not available)	Very high (when a sensor finds a neighbor it joins the network)	Based on neighbor discovery (time to reply or closest)

REFERENCES

[1] Woolston, K., and Albin, S., “The Design of Centralized Networks with Reliability and Availability Constraints”, Computers and Operations Research, Vol. 15, pp 207-217, 1988.

[2] Jaeook Lee and Sun Kang, “Satellite over Satellite (SOS) Network: A Novel Architecture for Satellite Network”, IEEE Infocom 2000, March 2000.

[3] Ganz, A.; Krishna, C.M.; Tang, D.; Haas, Z.J., "On optimal design of multitier wireless cellular systems," Communications Magazine, IEEE , vol.35, no.2, pp.88-93, Feb 1997.

[4] L.M. Feeney, A taxonomy for routing protocols in mobile ad hoc networks, October 1999, SICS Technical Report T99:07.

[5] J. N. Al-Karaki, A. E. Kamal, Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications Volume 11, Issue 6, pp. 6-28, Dec. 2004.

[6] Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y. Peer-to-Peer Based Multimedia Distribution Service, IEEE Transactions on Multimedia 6 (2) (2004).

[7] Wierzbicki, A., Strzelecki, R., Swierczewski, D. and Znojek, M. Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications, in: Second IEEE International Conference on Peer-to-Peer Computing (P2P2002), Linöping, Sweden, 2002.

[8] M. Jiang, J. Li, Y.C. Tay. “Cluster Based Routing Protocol (CBRP)”. Internet-draf, draft-ietf-manet-cbrp-spec-01.txt, National Uni-versity of Singapore, 14 August 1999.

[9] Chiang, C.C., Wu, H.K., Liu, W., Gerla, M., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", The Next Millennium, the IEEE SICON, April 14-17, 1997, Singapore, National University of Singapore, Kent Ridge (Singapore).

[10] Zygmunt J. Hass., Marc R. Pearlman, Prince Samar. “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, IETF, Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

[11] Z. J. Haas and M. R. Pearlman, “The zone routing protocol: A hybrid framework for routing in ad hoc networks,” in Ad Hoc Networks, C. E. Perkins, Ed. Reading, MA: Addison-Wesley, 2000.

[12] Mirco Musolesi, Cecilia Mascolo. “A Community Based Mobility Model for Ad Hoc Network Re-search”, Second International Workshop on Mul-ti-hop Ad hoc Networks: from theory to reality REALMAN 2006, 26 May 2006, Florence (Italy).

[13] P. F. Tsuchiya, “The Landmark Hierarchy: a new hierarchy for routing in very large networks,” In Computer Communication Review, vol.18 (Au-gust 1988), issue.4, pp. 35-42.

[14] Y. Rekhter, T. Li, “A Border Gateway Protocol 4 (BGP-4)”, RFC 1771, March 1995.

[15] G. Pei, M. Gerla and X. Hong, “LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility”, in Proceed-ings of IEEE/ACM MobiHOC, August 2000, pp. 11-18, Boston, MA (USA).

[16] J. M. McQuillan, I. Richer & E. C. Rosen. “The New Routing Algorithm for the ARPANET”. IEEE Trans. on Communications. Vol.28. Pp.711-719. 1980.

[17] Yogen K. Dalal y Robert M. Metcalfe, “Reverse path forwarding of broadcast packets”, Communications of the ACM. Volume 21, Issue 12 Pp: 1040 – 1048. December 1978

[18] R. Cohen and S. Havlin. Ultra small world in scale free graphs. Physical Review Letters, 90:058701, 2003.

[19] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the internet to random breakdowns. Physical Review Letters, 85:4626–4628, 2000.

[20] Gyorgy Hermann. Mathematical investigations in network properties. Proceedings IEEE Intelligent Engineering Systems, 2005. INES '05. Pp 79-82 September 16-19, 2005.

[21] X. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu, “A construction of locality-aware overlay network: mOverlay and its performance,” IEEE J. Select. Areas Commun., Special Issue on Recent Advances in Service Overlay Networks, vol. 22, pp. 18–28, Jan. 2004.

ARTICLE IN PRESS

Information Systems ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Information Systems

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

Multimedia group and inter-stream synchronization techniques: A comparative study

Fernando Boronat *, Jaime Lloret, Miguel García

Communications Department, Universidad Politécnica de Valencia, Escuela Politécnica Superior de Gandia—IGIC Institute Ctra. Nazaret-Oliva S/N, 46730 Grao de Gandia, Valencia, Spain

ARTICLE INFO

Article history:
Received 13 May 2008
Accepted 13 May 2008
Recommended by: D. Shasha

Keywords:
Synchronization
Inter-stream synchronization
Group synchronization
Multimedia systems

ABSTRACT

This paper presents the most comprehensive analysis and comparison of the most-known multimedia group and inter-stream synchronization approaches. Several types of multimedia synchronization are identified but only inter-stream and group synchronization algorithms are considered. This is the first survey including group synchronization techniques. A classification of the main synchronization techniques included in most of the analyzed algorithms complements the paper. Finally, a table is presented summarizing the main characteristics of each analyzed algorithm according to those techniques and other critical issues.

1. Introduction

In the past decade, we have seen a spectacular growth of the distributed multimedia real-time systems, most of them characterized by one or several sources transmitting (unicast or multicast) multimedia streams to one or several receivers, playing one or several of the streams. Multimedia systems are characterized by the computer-controlled integration of the generation, communication, processing and presentation of different media streams. These media streams can be divided into two categories: continuous and static (or non-continuous). Continuous media, e.g. video and audio, have well defined temporal relationships between subsequent ***Media Data Unit***s[1] **(MDUs)**. Static media, e.g. text, slides, images and graphics, have no temporal properties within themselves. Blakowski and Steinmetz [1] defined **multimedia system** as '*that system or application supporting the integrated processing of several types of objects or information, being at least one of them time-dependent*'.

A **distributed multimedia presentation** (**DMP**) integrates multiple media streams, e.g., audio, video, image, and text media, and possesses timeliness requirement of media units with respect to the presentation. DMP systems require flexibility and good quality of service (QoS) for multimedia data presentations. To ensure flexible and satisfactory presentations of multimedia data, collaborations between servers (sources), network and clients (receivers) must be carefully designed to retrieve the data from the disk (or database) and transfer the data to the client (receiver).

Due to the time dependency between the different media objects, a coordination and organization in time of the different information streams is needed. Such a process of maintaining the temporal relationship and guaranteeing a time-ordered presentation between the MDUs of one or several media streams is called the **multimedia synchronization process**. In this paper we use the **multimedia synchronization** concept to refer to the process of integration, in the presentation instant (*playout point*), of different types of media streams (continuous and/or static). Hereafter, we will use the term '*synchronization*' to refer to that concept. On the other

* Corresponding author. Tel.: +34 962 849 341; fax: +34 962 849 309.
E-mail addresses: fboronat@dcom.upv.es (F. Boronat), jlloret@dcom.upv.es (J. Lloret), migarpi@posgrado.upv.es (M. García).

[1] We have found that a MDU is also called *media or medium unit* (MU), *logical data unit* (LDU), *information unit* (IU) or *stream granules* (*unit of stream granularity*) by different authors.

0306-4379/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2008.05.001

Please cite this article as: F. Boronat, et al., Multimedia group and inter-stream synchronization techniques: A comparative study, Informat. Systems (2008), doi:10.1016/j.is.2008.05.001

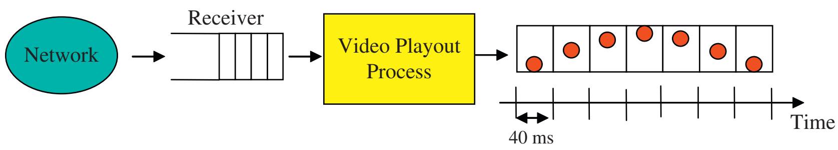


Fig. 1. Intra-stream synchronization.

hand, we use the term **multimedia synchronization algorithms** for protocols or solutions including the functionalities of maintaining the temporal relationships and guaranteeing a time-ordered presentation of one or several media streams. The solutions for synchronization use different techniques to coordinate the temporal and spatial object ordering, once the different media types have specific performance requirements.

Synchronization can be distinguished on different levels of abstraction. ***Event*[2]*-based synchronization*** must ensure a proper orchestration of the presentation of distributed multimedia objects. A multimedia object may be, for instance, a news cast consisting of several media objects, like audio and video. On a lower level, ***continuous synchronization*** or ***stream synchronization*** copes with the problem of synchronizing the playout of the data streams [2]. The classical example of stream synchronization is the synchronization between the audio stream and the associated lip movements in a speech (in this case a very accurate synchronization is required between both streams' playout processes), which is called ***lip-synchronization*** or ***lip-sync*** [3–5].

In continuous synchronization, we also can distinguish the temporal and MDUs relationships according to the use of either ***live*** or ***syntactic synchronization*** [1]. In the former case, the capturing and playback must be performed almost at the same time; in the latter case, samples are recorded, stored and played out at a later point of time. Live synchronization exactly reproduces the temporal relations made during the capturing process, while synthetic synchronization artificially specifies the temporal relations. Teleconferencing [6,7] is an example of a live synchronization application, while synthetic synchronization is often used in retrieval-based systems to rearrange multimedia objects to produce new combined multimedia objects (for example, video on demand (VoD) applications [5]). Temporal models allow specification of the temporal relations and operations of multimedia objects. These models can express complex operations by combining simple operations such as parallelized media streams, serialized multimedia objects and simple independent multimedia objects [1]. For live synchronization, the tolerable end-to-end delay is in the order of only a few hundred milliseconds. Consequently, the size of the elastic buffer must be kept small, and we can find trading-off requirements for jitter compensation against low delay for interactive applications. Synthetic synchronization of recorded media streams is easier to achieve than live synchronization: higher end-to-end

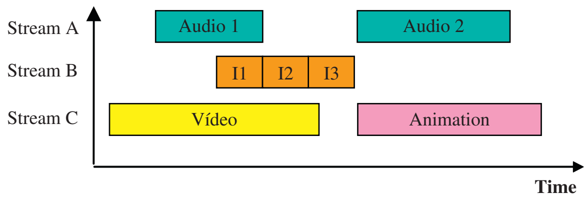


Fig. 2. Inter-stream synchronization.

delays are tolerable, and the fact that sources can be influenced proves to be very advantageous. For example, it is possible to adjust playback speed or to schedule the start-up times of streams as needed. However, as resources are limited, it is desirable for both kinds of synchronization to keep the required buffers as small as possible [8].

In temporal synchronization we can distinguish between intra-stream (or intra-media) synchronization, inter-stream (or inter-media) synchronization and group (or inter-destination) synchronization. The distinction between these three types of synchronization will allow us to identify the different techniques or mechanisms to achieve them.

Intra-stream synchronization deals with the maintenance, during the playout, of the temporal relationship within each time-dependent media stream, i.e. between the MDUs of the same stream. As an example, we can cite the temporal relationship between MDUs of a video sequence. If we suppose the video was captured at a generation rate of 25 frames/s, each frame has to be displayed during 40 ms in the visualization device. In Fig. 1, synchronization requirement is shown for a video sequence showing a jumping ball. When reaching the receiver, the MDUs will be stored in a reception buffer to be able to guarantee the intra-stream synchronization. It will be necessary to guarantee the existence of MDUs in the buffer during the playing process (avoiding buffer underflow situations—*starvation*) and/or to guarantee that the buffer is not full when new MDU arrive (avoiding buffer overflow situations—*flooding*). Moreover, the playout process should be able to consume the MDUs with the same appropriate rate.

On the other hand, **inter-stream synchronization** refers to the synchronization, during the playout, of the playout processes of different media streams (time-dependent or not) involved in the application. An example of the temporal relationships between media streams in a multimedia application is shown in Fig. 2 (display-time bar char of a presentation's temporal schedule). It shows a playout starting with a video, followed by an audio sequence and several static images (slides), and, when all these sequences finish, there is an animation with related

[2] An event is an occurrence in time that can be instantaneous or can occur over some time period [61].

audio comments. In this case, first, intra-stream synchronization is also needed, and then, during the playout, some actions to correct possible deviations between playout processes should be taken to guarantee inter-stream synchronization.

As examples of inter-stream synchronization, we can cite *lip-synchronization* (*lip-sync*, [3–5]) as explained above, the synchronization between static images (slides) and an audio sequence describing the slides, etc. For example, if a presentation of video and audio streams is conducted without enforcement of synchronization, jitter will gradually accumulate between both streams. Such jitter may severely affect the performance of the presentation, especially in applications where speech is involved. Thus, the synchronization of multiple media streams becomes an essential prerequisite to any successful multimedia presentation application.

Manvi and Venkataram [9], present an inter-stream synchronization classification with three types: *point*, *real-time continuous* and *adaptive synchronization*. Point synchronization realizes that the start/completion time of the MDUs of the streams is synchronized with a certain specified synchronization point between the streams. In real-time continuous synchronization, MDUs are synchronized with a real-time axis. In adaptive synchronization, for example, the presentation time of the MDUs can be adjusted at regular intervals with change in network delays to reduce the losses.

From this point of view, we can also define a ***multimedia presentation*** as a set of media streams upon which synchronization constraints are specified on the display operations to enforce both intra and inter-stream constraints.

Apart from the above types of synchronization, in multicast communications, we can find another type of synchronization, called **group or inter-destination synchronization**, involving the synchronization of the playout processes of different streams in different receivers, at the same time, to achieve fairness among the receivers. We can cite the example of teleteaching applications in which a teacher could send (multicast) a video sequence (documentary or film—stored content stream) and, during the session, sometimes the teacher could make occasional comments about the video (live content stream). Network quizzes are other examples, in which the same multimedia question must appear at the same time to all the participants to guarantee fair play. In the first example, a simultaneous playout of the streams is important for both stored content and live content streams. Even if we only send the video stream (documentary or film), each video MDU (frame) should be played simultaneously in all the receivers (students) and then the students could comment the video content with other students. To guarantee that the initial playout instant (the playout beginning or starting point) should be the same for all the receivers. Once the playout processes have started simultaneously in each receiver, the temporal relationships between MDUs of the same stream should be maintained by the intra-stream synchronization process for that media stream. Nevertheless, due to the difference between end-to-end delays (due to different network delays, different consumption rates of the different receivers, etc.), resynchronization processes will be needed to maintain the receivers synchronized (group synchronization). In Fig. 3, we can see the playout of the above jumping ball sequence, synchronized in all the receivers.

In [10–16] another kind of synchronization (and solutions for it) is presented (***interactive synchronization***) for interactive distributed multimedia applications (IDMP). In some of these applications, VCR-like user interactions allow users to modify the presentation configuration at any time during the presentation. Typical user interactions include *fast forward (FF)*, *reverse (RR)*, *skip* and *pause/restart* the playout. There is a special difficulty in providing such user interactions due to the fact that they are issued dynamically and unpredictably during the presentation, which complicates the synchronization control.

The maintenance of temporal relationships within a stream or among the multimedia streams usually depends

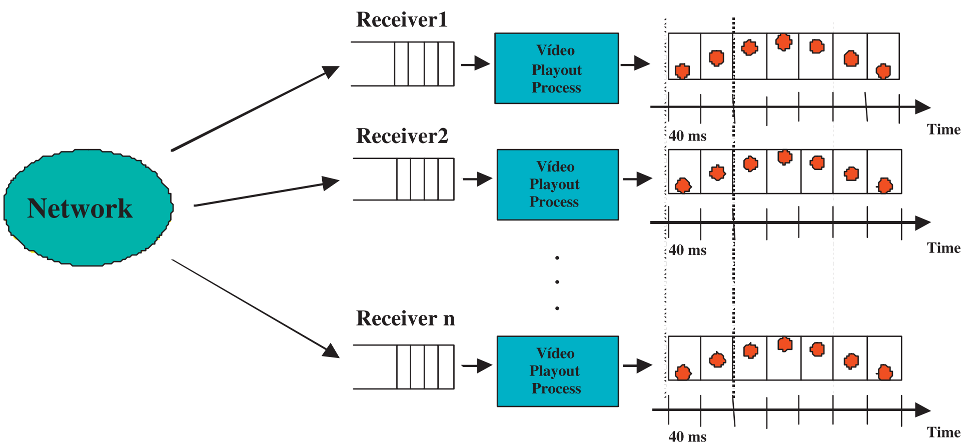


Fig. 3. Group synchronization.

on the following parameters, which can be tackled either individually or in an integrated manner [9]:

- *Network delays*: The delays experienced by the MDUs in the network to reach its receiver, which varies according to network load.
- *Network jitter*: It denotes the varying delay that stream packets experience on their way from the sender to the receiver network I/O device. It is introduced by buffering in intermediate nodes. It refers to the delay variations of inter-arrival of packets at the receiver because of varying network load.
- *End-system jitters*: Delay variations in presentation at the receiver because of varying workstation load and protocol processing delays. It refers to the variable delays arising within the end-systems, and is caused by varying system load and the packetizing and depacketizing of MDUs with variable size, which are passed through the different protocol layers.
- *Clock skew*: The clock time difference between the sender and the receiver.
- *Clock drift*: The rate of change of clock skew because of temperature differences or imperfections in crystal clocks.
- *Rate drift*: Change in generation and presentation rates because of server and receiver load variations.
- *Network skew*: Time difference in arrival of temporally related packets of streams, which is a differential delay among the streams.
- *Presentation skew*: Time interval in which the temporally related packets of the streams are presented.

Jitter is commonly equalized by the use of elastic buffers at the receivers (there are lots of buffering techniques).

Capture, reproduction and presentation of continuous media is driven by end-system clocks, but due to temperature differences or imperfections in the crystal clock, their frequencies can differ over a long period of time. The result is an offset in frequency to real time and to other clocks, which causes rate drifts. The problem of clock drift can be coped with by using time-synchronizing protocols within a network (for example, using the Network Time Protocol (NTP) or Global Positioning System (GPS) devices). Otherwise, if the problem of clock drift is neglected, buffer overflow (*flooding*) or buffer underflow (*starvation*) at the receiver will appear over a long period of time. The effect of clock drift is also known as *skew*, which is defined as an average jitter over a time interval.

Usually, networks are dynamic and have changing network conditions, not introduced by jitter, which are referred to a variation of connection properties (for example, an alteration of the average delay or an increasing rate of lost MDUs). These effects strongly depend on the QoS the underlying network can provide.

Apart from the network delay, MDUs also experience delay due to packetizing/depacketizing, the processing through the lower protocol layers, and the buffering at the transmitter and receiver sites. Fig. 4 shows the cumulative delay of two independent streams whose generation starts at the same time. Due to the different delays, the MDUs of the two streams (generated at the same time) would not be played out simultaneously if no synchronization techniques were used.

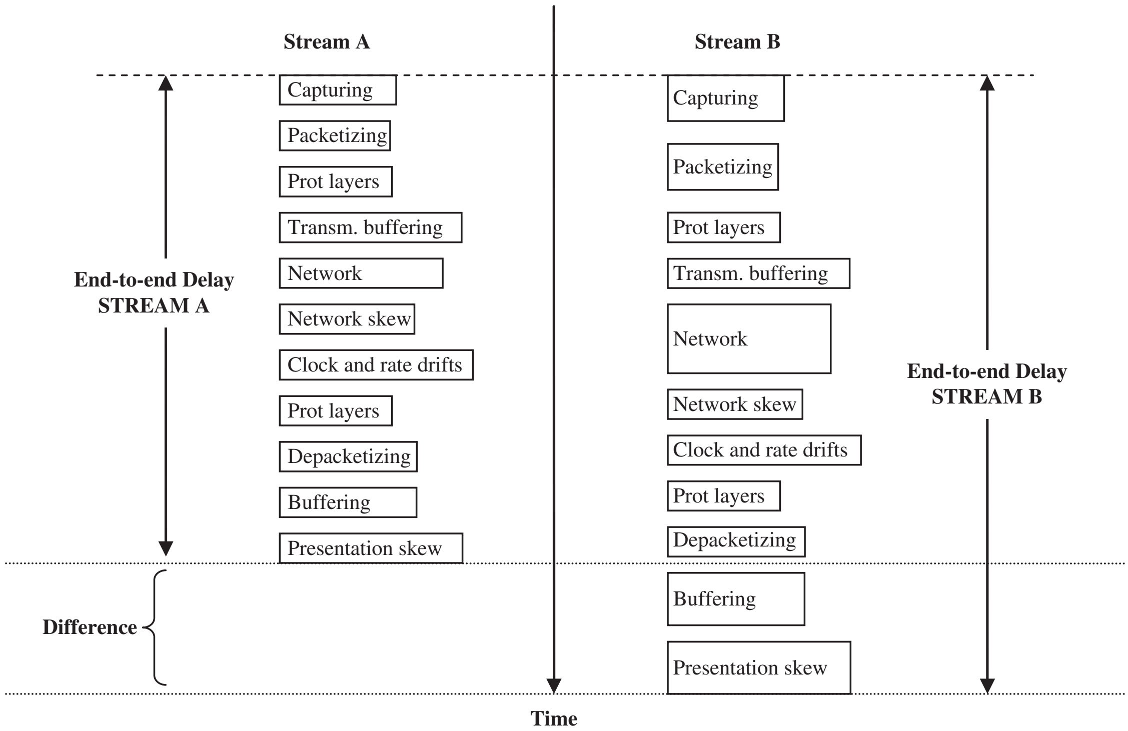


Fig. 4. Delays experienced by two independent media streams [8].

Synchronization mechanisms are needed to cope-up with all the above problems to ensure the time ordering of the stream/s and to maintain the presentation quality. Furthermore, the synchronization mechanism has to be adaptive regarding changes of the network conditions and to source drop-outs, which are a realistic assumption when using non-real-time operating systems.

There are lots of intra-stream synchronization solutions most of which try to avoid receiver buffer underflow and overflow problems. In [17], a comparative survey between intra-stream synchronization techniques can be found.

Over the last few years, new techniques have been developed to improve synchronization in multimedia systems, such as fixed and mobile agent-based techniques [9,18] and aspect-oriented programming-based techniques [19].

This paper is only focused on multimedia group and inter-stream synchronization techniques. We present the basic characteristics of them and analyze and compare qualitatively many of the solutions proposed in the past.

The rest of the paper is organized as follows. In the next section several past classifications found in the literature are presented which represent the basis of our classification. In Section 3, we present, as an example to make the understanding of the paper easier, a proposal developed by the authors to guarantee group and inter-stream synchronization using standard protocols. Section 4 presents the description and classification of the most common techniques for synchronization used by the studied solutions. In Section 5, a comparison table between those solutions (chronologically ordered, to indicate their appearance in time) is shown, according to several critical issues, very related to multimedia group and inter-stream synchronization. Finally, the paper ends by presenting our conclusions and the references.

2. Background

As far as we know, a common classification scheme for synchronization approaches does not exist. Most of the studied synchronization mechanisms are either application-specific or try to cover synchronization on a more abstract level independent of the application at hand. Surveys of multimedia synchronization mechanisms can be found in [8,16,20–23].

For this study, we have consulted the patterns in those references, followed by some authors in the past to compare multimedia synchronization techniques. In this work and henceforth the relation between algorithm/solution and techniques is the following: an algorithm can involve several control techniques.

Ehley et al. [21] present an early classification of the multimedia synchronization algorithms, grouping them in two synchronization schemes: distributed (in a network environment) and local (inside a workstation). Köhler and Müller [16] present another classification based on several factors: clocks (globally synchronized clocks and locally available clocks), synchronization mechanism location (it can be performed either at the sender or the receiver of continuous media information, but sender control requires some kind of feedback) and synchronization control techniques (only two are considered: clock frequency adjustment and skip and/or pause actions).

In [8,16], only three classification criteria (time, location and method) are used to summarize some solutions, which are systematized graphically in a 3D cube, with each criterion in a different axis. The solution space for playout synchronization consists of three almost orthogonal design criteria with two main choices in each dimension. The first decision is, whether the systems have an explicit common understanding of time or not. In the former case, some kind of clock synchronization takes place. The presentation time of a MDU can be calculated from an absolute or relative timestamp carried with every unit. If no clock synchronization takes place, playout synchronization can be achieved based on buffer control mechanisms. The second criterion is the location of synchronization actions (source or receiver). The third dimension distinguishes the methods that are used to correct asynchrony. Restoring synchronization could be done either by speeding up or slowing down the presentation or generation of MDUs, or by *stuffing*—well known method from bit or byte synchronization which means duplicating/deleting MDUs or pausing/skipping MDUs, respectively)

Perez-Luque and Little [23] develop a uniform, theoretical foundation for discussing multimedia synchronization and temporal specification. A temporal reference framework is proposed, focused on time models and it is used to compare some existing temporal specification schemes and their relationships to multimedia synchronization. That analysis explains why there are so many synchronization frameworks, how a multimedia scenario can be represented with different temporal specification schemes, and why some specification schemes cannot model all scenarios.

The classification presented by Ishibashi and Tasaka [22] is the most comprehensive one we have found and the one we have used as our starting point. In it only the intra-stream and inter-stream techniques used in each analyzed solution were taken into account, but address only 1:1 and n:1 communication.

In our comparative study, we have also included solutions for communications for 1:n and n:m communication and added new inter-stream synchronization solutions (previous (not considered in [22]) and later ones), such as those presented in [8,24–29,30,31] (some of them developed by the authors of this paper) [2,3,5,9,10–15,19,32–68]; and new versions of algorithms included in the original comparative survey, such as the ones in [5,38,69,70–75]. For example, in [69,72], the synchronization maestro scheme (SMS) for group synchronization, employed together with the virtual time rendering (VTR) media synchronization algorithm [76] has been enhanced so that the SMS scheme can be used efficiently in a P2P-based system and in a networked real-time game with collaborative work, respectively. Likewise, in [77], the scheme SMS and the distributed control scheme (DSC, defined in [47]), both used for group synchronization, have also been enhanced, by taking into account the importance of the media objects.

In [20], the authors present (in Spanish language) a preliminary and shorter survey with 28 studied solutions. In this paper, we have completed and updated that survey with many new synchronization solutions (up to 53) and with new techniques and parameters to compare. In [22], we found eight algorithms that only included intra-stream synchronization. Those solutions and some others, which the authors have not been able to get, have not been included in this new study. Moreover, as an additional contribution, we have added other factors to the original comparative survey that we consider very relevant in the classification (for example, whether or not the solution uses a new specific protocol with new control messages or whether it uses a standard protocol, such as RTP/RTCP; whether it uses feedback from the receivers; the included group synchronization techniques, etc.). Another contribution has been the inclusion of the group synchronization approaches in the study (not included in [22]), such as the ones presented in [10,11,14,15,26,30,31,41–48,53,61, 67–69,72,78]).

One of the main aims of the paper is to present only a qualitative comparative study, and not a quantitative comparative one, because the relations between the solutions we have found are not clear enough. One of the main reason is that the situations and environments those solutions have been developed for are very different. Moreover, no standard measurements have been defined to evaluate objectively the multimedia synchronization performance of the techniques. Nowadays, the quantitative relationships between the solutions seem quite chaotic.

3. Example of group and inter-stream synchronization algorithm

In this section, our proposal for multimedia group and inter-stream synchronization is qualitatively described as an example. More details regarding our solution and other components can be found in [30,31]. It is based on two previous protocols: the *Feedback Protocol* [79–84] and the *Feedback Global Protocol* [38]. The proposal, which uses the existing RTP/RTCP protocol suite (already used in most of the current multimedia applications for data transport and control), supposes the intra-stream synchronization is guaranteed by some technique (for example, one of the solutions classified in [17]) maintaining a correct and continuous playout of each multimedia stream (regardless of its nature), and includes two synchronization processes (group and inter-stream):

(1) *Group Synchronization*. This process has two phases: an *initial phase* in which all the receivers should start the playout of one of the streams (considered as the *master stream*), at the same instant (*Initial Playout Instant*); and then, the *second phase*, in which all the receivers should play that stream synchronously as continuously as possible.
(2) *Inter-stream Synchronization*. Inside each receiver there is an internal or local synchronization process between all the playout processes of all the streams the receiver is playing. This process is also called *multimedia fine inter-stream synchronization*. The playout processes of the other streams (considered as *slave*) will synchronize to the one of the master stream (also involved in the group synchronization process). This process will maintain the temporal relationships (the same as the ones existing at the generation time) between streams during the playout in each receiver.

A global time reference exists in all the sources and receivers (acquired, for example, using the NTP or GPS devices).

The set up is shown in Fig. 5. As it looks a bit confusing, we have divided it into several figures to make it clearer (Figs. 6–9).

In Fig. 6, we can appreciate the existence of a transmission (*multicast* or *unicast*), of several multimedia streams, using RTP (for data transmission) and RTCP (for control and feedback information transmission), from one or several sources to one or several receivers (our solution is general and covers the case in which a unique source sends multiple streams to one or several receivers). One of the streams has been chosen as the *master stream* (thicker lines and arrows) and, moreover, we have selected one receiver as the *master receiver* (dark in the figure), whose master stream playout point in every moment will be taken as the reference to determine the state (advanced or delayed form that one) of the playout of the other receivers (*slave receivers*). The master receiver can be selected taking into account several criteria beyond the scope of this paper [30,31].

Taking into account the RTCP feedback messages, the proposed synchronization algorithm, which will be executed at the master stream source (called *Synchronizer Source*), will use the feedback RTCP RR packets [85] which are extended [85] to include information useful for our algorithm, and new defined RTCP APP packets [85], used to determine and communicate the playout state of the master stream, both in and to all the receivers, respectively. In this way, the Synchronizer Source will be able to know the state of the playout processes during the session. This process is shown in Fig. 7.

The Synchronizer Source, when deviations (asynchronies) are detected (receivers whose master stream playout process is either advanced or delayed with respect to the master receiver's process) exceeding a threshold value, will generate action messages to make them correct the master stream playout, by skipping or pausing MDUs in the playout processes. This way the synchronized master stream playout in all the receivers (group synchronization, between receivers) is guaranteed. This process is shown in Fig. 8.

Once the group synchronization between receivers has been achieved regarding the master stream playout, the proposal also has to guarantee the local inter-stream synchronization (for example, *lip-sync*, [3–5]). For this purpose, an internal inter-processes communication channel can be used (for example, *mbus* [86], also used in [28]). In each receiver, and locally, the playout process of the master stream (synchronized with all the other

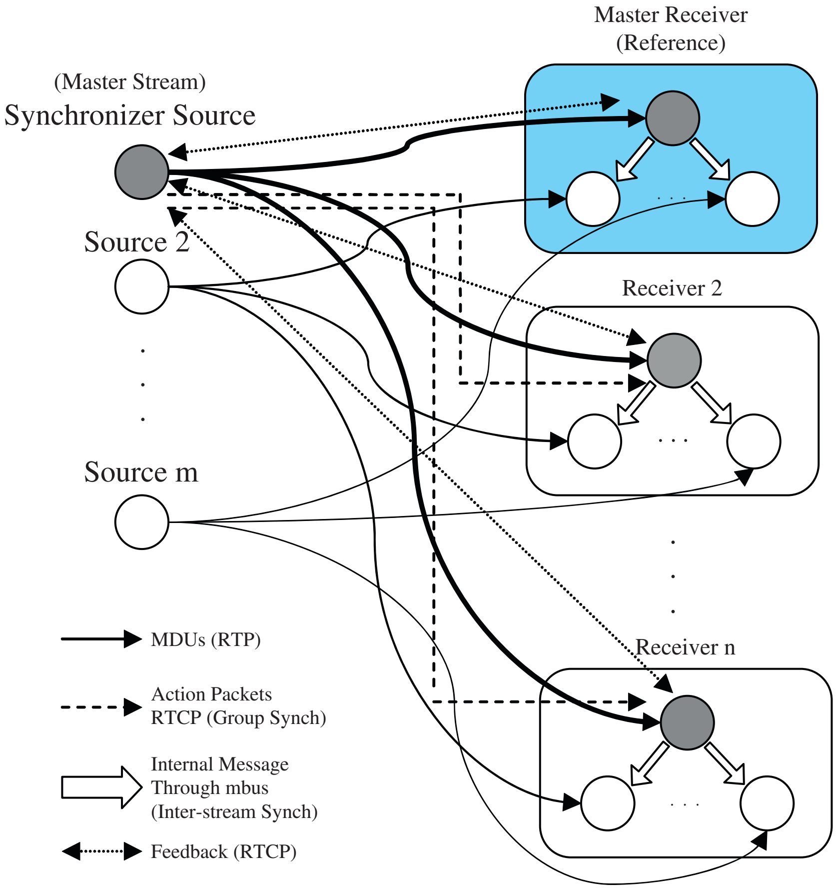


Fig. 5. Data and control streams.

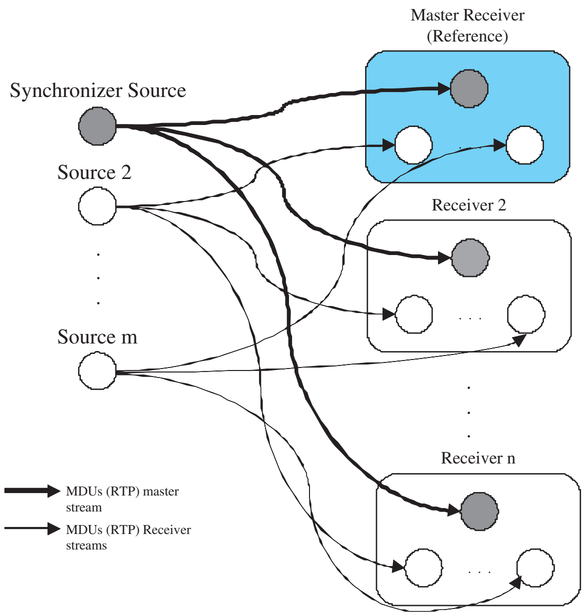


Fig. 6. Stream data transmission using RTP.

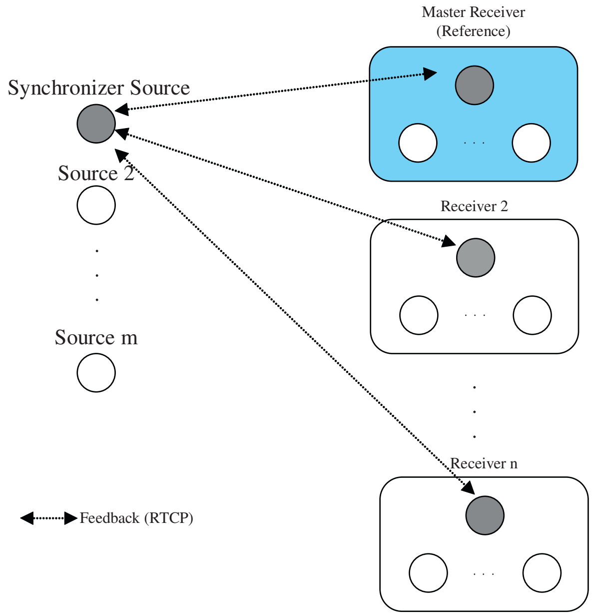


Fig. 7. Feedback messages (only for master stream).

Please cite this article as: F. Boronat, et al., Multimedia group and inter-stream synchronization techniques: A comparative study, Informat. Systems (2008), doi:10.1016/j.is.2008.05.001

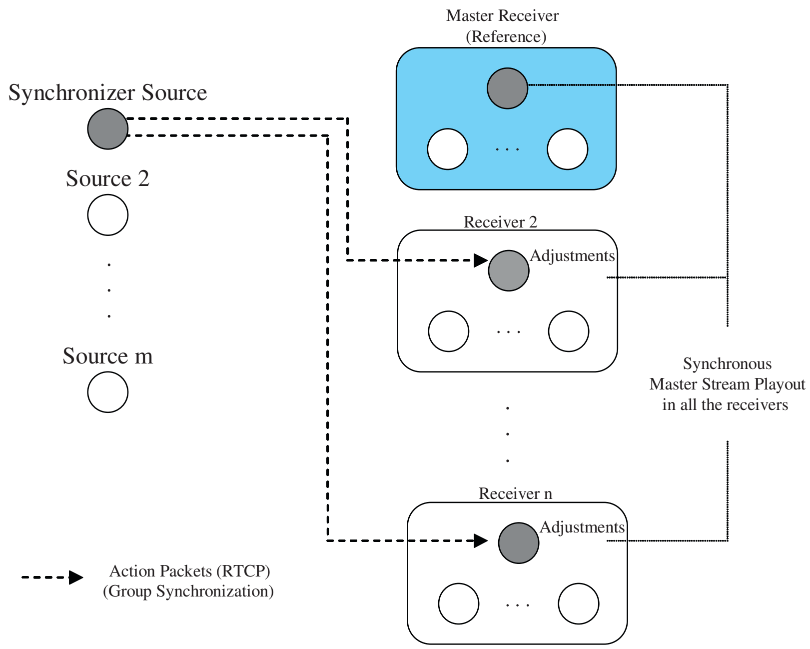


Fig. 8. Action RTCP messages sent by the Synchronizer Source to the slave receivers.

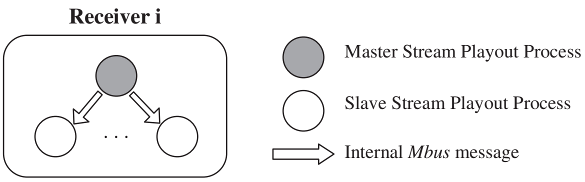


Fig. 9. Local inter-stream synchronization.

receivers using the above group synchronization process) will send its playout state periodically to the other slave streams' playout processes, to make them synchronize their playout states with synchronization actions, such as skips or pauses. This process is depicted in Fig. 9.

4. Synchronization techniques

Here, we present the main synchronization techniques we have found in the most representative solutions we have studied, classified into several categories. As several techniques can be included in a specific solution, each technique should be unique and indivisible (atomic), i.e. with no different functions from the multimedia synchronization point of view.

These techniques are summarized in Table 1. In Table 2, the main characteristics of all the studied synchronization solutions are presented, including the techniques each solution uses (right column).

The names of the techniques are presented in italics.

4.1. Specific techniques for group synchronization

Three of the most common techniques for group synchronization are: the *master/slave receiver scheme*, the *synchronization maestro scheme (SMS)* and the *distributed control scheme (DCS)*.

In the master–slave receiver scheme (initially presented in [78]), receivers are classified into a master receiver and slave receivers. None of the slave receivers send any feedback information about the timing of the playout processes. It adjusts the playout timing of MDUs to that of the master receiver. Only the master receiver sends (multicast) its playout timing to all the other (slave) receivers.

The SMS (initially presented in [42]) is based on the existence of a synchronization maestro (it can be the source or one of the receivers) which gathers the information on the playout processes from all the receivers and corrects the playout timing among the receivers by distributing control packets. In order to do this, each receiver sends (unicast) the information to the maestro, and the maestro sends (multicast) the corrected playout timing. The author's solution, presented in Section 3, follows this scheme.

In the distributed control scheme (initially presented in [47]), all the receivers can exchange (multicast) the control packets or use timestamps in media packets to calculate playout delays to achieve group synchronization. Each receiver decides the reference playout timing from among the output timing of itself and that of the other receivers. In [47,48], a distributed control scheme is proposed, which adaptively keeps the temporal and causal relationships according to the network load under distributed control. For group synchronization, the scheme adopts a group synchronization algorithm, which is called the *distributed control scheme*. In [11], a bucket synchronization mechanism is presented. In [53], the use of a local-lag and a timewarp algorithm is proposed to avoid inconsistencies between users in a replicated consinuous application (e.g., a network game application).

Table 1
Synchronization techniques

Location	Technique	Technique's purpose
Basic Control	Source control	Add information useful for synchronization (timestamps, sequence numbers (identifiers) event information and/or source identifiers.)
	Receiver control	Buffering techniques
Preventive Control	Source control	Initial playout instant calculation Deadline-based transmission scheduling Interleave MDUs of different media streams in only one transport stream
	Receiver control	Preventive skips of MDUs (eliminations or discardings) and/or preventive pauses of MDUs (repetitions, insertions or stops) Change the buffering waiting time of the MDUs Enlarge or shorten the silence periods of the streams
Reactive Control	Source control	Adjust the transmission timing Decrease the number of media streams transmitted Drop low-priority MDUs
	Receiver control	Reactive skips (eliminations or discardings) and/or reactive pauses (repetitions, insertions or stops) Make playout duration extensions or reductions (playout rate adjustments) Use of a virtual time with contractions or expansions Master/slave scheme (switching or not) Late event discarding (Event-based) Rollback techniques (Event-based)
Common Control	Source control	Skip or pause MDUs in the transmission process Advance the transmission timing dynamically Adjust the input rate Media Scaling
	Receiver control	Adjust the playout rate Data interpolation

We can also classify the above schemes into centralized and distributed control schemes. The former schemes have their own advantages and disadvantages. For instance, they can more easily preserve causality and there is less possibility of inconsistency of state among the session members (receivers) occurring than with distributed control schemes. However, they have larger network delays, lower reliability and poorer scalability. Therefore, distributed control is also desirable for causality and media synchronization. In [47], the advantages and disadvantages of the *SMS* and the *distributed control scheme* are discussed in terms of reliability, control speed, control overhead, etc. In [77] both schemes for group synchronization have been enhanced by taking into account the importance of the media objects, for its application in networked virtual environments. In that work, the concept of *global importance* (importance which is judged from a point of view of all the users (receivers) in the virtual environment) is introduced in addition to the local importance (importance which is judged from a viewpoint of each user and used to change the intra-stream and inter-stream synchronization accuracy).

In [69,72] the SMS for group synchronization, employed together with the VTR media synchronization algorithm [76] has been enhanced so that that scheme can be used efficiently in a P2P-based system and in a networked real-time game with collaborative work, respectively. Likewise, in [77] the SMS and the DSC schemes, both used for group synchronization, also have been enhanced, by taking into account the importance of the media objects.

In [87], the three above schemes, based on the VTR algorithm, have been evaluated in a quite simple Multicast Mobile Ad Hoc Network (MMAHN).

4.2. Generic synchronization techniques

Although many ways to classify the synchronization techniques can be found, we have chosen, and extended, the ones described by Ishibashi and Tasaka [22] and Liu [50], based on each techniques' purpose and the locations at which they are employed. We classify the synchronization techniques into four groups:

(a) *Basic control techniques*, needed in most of the solutions and essential to preserve the temporal structures.
(b) *Preventive control techniques*, needed to avoid the asynchrony (situation of out of synchrony), before it appears.
(c) *Reactive control techniques*, needed to recover from asynchrony after it has been detected.
(d) *Common control techniques*, which can be used for both prevent (prevention) and/or correct (reaction) situations of asynchrony.

Any one of these techniques, either alone or in combination with others, can be employed to achieve the desired synchronization for a targeted application. In most of the solutions, the Basic Control techniques are usually complemented, at the same time, with preventive and/or reactive control techniques. The preventive control techniques cannot usually avoid completely the appearance of asynchrony, so the combination with reactive control techniques is also needed.

Generally, the above synchronization techniques can be applied at the source (multimedia server) and/or the receiver. On the one hand, some control techniques are always needed at the receiver side because of the existence of network jitter. On the other hand, control techniques at the source side need *feedback* information from the receivers or from the network to let the source/s know the synchronization and/or network QoS in each moment to proceed consequently. In some techniques, source/s and receivers cooperate to control the synchronization processes. We divide the above four techniques into two groups *according to their location* (source or receiver), as in [16]. In some solutions for mobile

Table 2
Multimedia inter-stream and group synchronization solutions

NAME	Clock	Delay limits	MDU generation	Stored or live contents	Intra, inter and group synchronization	Group syncronization techniques	*Master/Slave (M/S)*	*Feedback utilization*	Synchr. information	Location	RTP use	Synchronization techniques
ACME [93]	Global	–	Periodical	Both	Intra and inter	–	M/S streams and without relationship	–	Timestamps	Receiver	No	Reactive skips and pauses Virtual time expansion Transmission rate adjustments Playout rate adjustments Data interpolation
[91,92]	Global or local	–	Periodical	Stored	Inter	–	Without relationship	Yes (but not used for synch.)	Seq. number	Receiver	No	Decreasing the number of media streams Preventive and reactive skips (discardings) and pauses (duplicates) to change the playout rate MDU insertions
[95]	Global	Known	Periodical	Live	Inter	–	Without relationship	No	Seq. number	Source and receiver	No	Decreasing the number of media streams Reactive skips and pauses
FP [79–84]	Local	Known	Periodical	Stored	Intra and inter	–	M/S streams	Yes	Timestamps Seq. number	Source and receiver	No	Skips and pauses at source side
MCP [67,68]	Global	–	Periodical and no Periodical		Inter and group	Token-based	–	No	Timestamps Seq. number context information	Receiver	No	Skips and pauses at receiver side MDU discarding
[18,57,58]	Local	Unknown	Periodical and no Periodical	Stored	Intra and inter	–	Without relationship	No	Seq. number	Source and receiver	No	Initial transmission and playout instants Deadline-based transmission scheduling Insertion of synchronization points (coarse synchronization) Reactive skips and pauses
Fsp [36]	Global	Unknown	Periodical and no Periodical	Both	Inter	–	–	No	Timestamps	Receiver	No	Playout duration extensions or reductions
VTR [70,71,73,74,76,78,90]	Global and local	Unknown	Periodical and no Periodical	Both	Intra and inter and group [78]	M/S receiver scheme [78]	M/S streams	Yes	Timestamps Seq. number	Source and receiver	In [70,71,74]	Change of the buffering time according to the delay estimation Decreasing the number of media streams Preventive pauses Reactive skips and pauses Skips at the source side Playout duration extensions or reductions Virtual local time expansions or contractions Media scaling Interleaving MDUs [73,74]

ASP [2,62]	Global	–	–	Both	Intra and inter	–	M/S streams	–	Timestamps	Receiver	No	Initial transmission and playout instant Reactive skips and pauses (duplicates) Change of the buffering time according to the delay estimation Playout duration extensions or reductions Master/slave switching Playout rate adjustment
Concord Algorithm [63]	Global and local	Known	Periodical –	–	Intra and inter	–	M/S streams and without relationship	–	Timestamp in 1st packet or especial packets	–	No	MDU discarding Pauses in the slave stream playout
[94]	Local	Unknown	–	Both	Intra and inter	–	M/S streams	Yes	Timestamps Seq. number	Source and receiver	No	Transmission and playout rate adjustments Skips and pauses at source site Reactive skips and pauses Playout duration extensions or reductions Late MDU discarding
[26]	Global	Known	–	–	Intra, inter and group	Distributed control scheme	M/S streams M/S receivers (chairman)	–	Timestamp in 1st packet	–	No	Initial transmission and playout instant Playout rate adjustments (receiver's clock) Master/slave receiver swhitching (chairman)
[27]	–	Known	–	Stored	–	–	Without relationship	No	Playout deadline	Source	No	Deadline-based transmission scheduling
[8,29]	Local	Unknown	–	Stored	Intra and inter	–	Without relationship	Yes	Timestamps Seq. number	Source and receiver	Optional.	Initial transmission instant synchronization to achieve, subsequently, the initial playout instant synchronization Playout duration extensions Preventive skips and pauses at source side
MultiSynch [96]	–	–	Periodical	–	Intra and inter	–	Without relationship	–	Timestamps	Receiver	No	Reactive skips (discardings) and pauses Low priority MDU dropping
[4]	Global	Unknown	Periodical	Live	Intra and inter	–	M/S streams and without relationship	–	Timestamps	Receiver	Yes	Master/slave switching Playout duration extensions or reductions
SSP [89]	Local	Known	Periodical and no Periodical	Stored	–	–	–	Only at the initial stage	Timestamps	Source and receiver	No	Initial playout instant Deadline-based transmission scheduling Low priority MDU dropping Preventive skips (selective discardings) and pauses (duplicates) or deliberated delays Virtual time expansion

Table 2 (*continued*)

NAME	Clock	Delay limits	MDU generation	Stored or live contents	Intra, inter and group synchronization	Group syncronization techniques	*Master/Slave (M/S)*	*Feedback utilization*	Synchr. information	Location	RTP use	Synchronization techniques
[6]	Local	Unknown	Periodical and no Periodical	Without distinction	Intra and inter	–	Without relationship	–	Timestamps	Receiver	No	Reactive skips (discarding) and pauses
[99]	Global	Known	Periodical	Stored	Intra and inter	–	M/S streams and without relationship	–	Timestamps (in some MDUs)	Source and receiver	No	Reactive skips and pauses at source and/or receiver side
[64]	Local	Unknown	–	Both	Inter	–	–	Yes	Timestamps	Source (Synchronization Server)	No	Skips and pauses at source side
MSTP [100]	Local	Unknown	Periodical and no Periodical	Both	Inter	–	M/S streams	–	Seq. number. Seq. number of the master stream MDU to synchronize with	Receiver	No	Reactive skips and pauses in slave streams' playout Slave streams MDU discarding Late MDU discarding
[101]	Local	Unknown (maximum Jitter)	Periodical	–	Inter	–	Without relationship.	Yes	Timestamps Seq. number	Source	No	Skips and pauses (noise MDU insertion) at source side
[98]	Local	Known	Periodical	Both	Intra and inter	–	M/S streams	No	–	Receiver	No	Reactive skips and pauses Master/slave switching
SMS [14,15,41–46,72]	Global	Known	Periodical and no Periodical.	Both ([78] for stored and [41] for live)	Group (VTR is used for Intra and inter)	Synchronization maestro scheme	M/S streams M/S receivers	Yes	Timestamps Seq. number	Receiver	No	VTR [76] techniques Initial transmission instant (only in [42])
[5,60]	Local	–	Periodical	Stored	Inter	–	M/S streams	Yes	Timestamps	Source and receiver	No	Initial transmission rate adjustment Playout rate adjustments Skips and pauses Slave stream's MDU discarding
[12–14]	Local	Unknown	Periodical and no Periodical	Stored	Intra and inter	–	M/S streams	Yes	Seq. number	Receiver	No	Initial playout instant Late and useless/redundant MDU discarding Reactive skips and pauses in slave streams' playout Master/slave switching
[97]	Local	Unknown	Periodical	Both	Intra and inter	–	M/S streams	Yes	Seq. number	Receiver	No	Playout rate adjustments Playout duration extensions or reductions Reactive skips (discardings) and pauses (duplicates)
[88]	–	–	Periodical	Both	Intra and inter	–	M/S streams	No	Seq. number Synchr. marker	Receiver	No	Reactive skips and pauses Playout rate adjustments

BS [11]	Global	Known	Periodical and no Periodical	Live	Inter and group	Distributed control scheme	Without relationship	No	Timestamps Seq. number	Receiver	Yes	Skips (discardings) and pauses (duplicates) Late events are dropped
[54–56]	Global	Unknown	Periodical and no Periodical	Stored	Intra and inter	–	–	Yes	Timestamps Seq. number	Source and receiver	Yes	Transmission (or playout) rate adjustments Low priority MDU dropping. Late MDU discarding. Playout rate adjustments. Dynamic playout scheduling
[7]	Local	Unknown	Periodical and no Periodical	Live	Intra and inter	–	M/S streams	No	Timestamps	Receiver	No	Reactive skips and pauses Virtual time expansion and contraction Master/slave switching
DCS [47,48]	Local	Unknown	Periodical and no periodical	Both	Group (VTR is used for Intra and inter)	Distributed control scheme	M/S streams	No (Timing information exchange between receivers)	Timestamps	Receiver	No	VTR [76] techniques Playout rate adjustments
PARK [39,40]	Local	Unknown	Periodical and no periodical	Stored	Intra and inter	–	M/S streams	Yes	Seq. number	Source and receiver	No	Initial transmission instant Transmission rate adjustments Late slave streams' MDU discarding Reactive skips and pauses in slave streams' playout Master/slave switching
[25]	–	Unknown	Periodical and no periodical	Stored	Intra and inter	–	–	No	–	Source	–	Deadline-based transmission scheduling Initial transmission instant Transmission rate adjustments. Low priority MDU dropping (by source)
[37]	Global or local	Known	Periodical and no periodical	Live	Intra and inter	–	–	Only at the initial stage	–	Receiver	No	MDU discarding Playout duration extensions or reductions (reduction of the silent periods duration and 'Gap' insertion)
RVTR [75]	Global	Unknown	Periodical and no periodical	Both	Intra and inter	–	M/S streams	Yes	Timestamps Seq. number	Source (retrans.) and receiver	Yes	VTR [76] techniques Playout duration extension or reduction
FGP [38]	Global	Known	Periodical	Both	Intra and inter	–	M/S streams	Yes	Timestamps Seq. number	Receiver	No	Reactive skips and pauses
[49]	Global	Unknown	Periodical	Live	Intra and inter	–	–	Yes	Timestamps Seq. number	Receiver	Yes	Playout rate adjustments

Table 2 (*continued*)

NAME	Clock	Delay limits	MDU generation	Stored or live contents	Intra, inter and group synchronization	Group syncronization techniques	*Master/Slave (M/S)*	*Feedback utilization*	Synchr. information	Location	RTP use	Synchronization techniques
RTP-FGP [30,31]	Global	Unknown	Periodical and no periodical	Both	Inter and group	Synchronization maestro scheme	M/S streams M/S receivers	Yes	Timestamps Source id.	Source and receiver	Yes	Initial playout instant Reactive skips and pauses at the receiver side Playout rate adjustment Virtual time expansion Master/slave receiver switching (group synchronization)
ALPSMS [66]	Global	Unknown	Periodical and no periodical	Both	Inter	–	Without relationship	–	Timestamps Seq. number	Receiver	No	Change of the buffering time according to the delay estimation Late MDU discarding
[35]	–	Unknown	Periodical and no periodical	Both	Inter	–	M/S streams	Yes	Event number Timestamps	Source and receiver	–	Event-based synchronization control (no time-based control is used) Late MDU discarding
MoSync [32–34]	Local	Unknown (but Jitter bounded)	Periodical and no periodical	Both	Intra and inter	–	Without relationship	Yes	Timestamps Seq. number server/ source number	Source, network base station and mobile receiver	–	Initial transmission instant Deadline-based transmission scheduling (achieved by adopting various transfer rates) Dynamic playout scheduling
[50–52]	Local	Unknown	Periodical	Live	Intra and inter	–	M/S streams	No	Timestamps	Receiver	Yes	Virtual local time expansions or contractions MDU playout duration extensions or reductions Reactive skips
LSA [3]	–	–	Periodical	Both	Inter	–	M/S streams	No	Timestamps	Receiver	–	Playout rate adjustments
LL-TW [53]	Global	Unknown	Periodical and no periodical	Live	Group	Distributed control scheme	–	No	Timestamps	Receiver	–	Event-based synchronization control Playout duration extension Rollback-based techniques
[19]	Global	Unknown	Periodical and no periodical	Both	Intra and inter	–	–	No	Timestamps Seq. number	Receiver	Yes	Dynamic playout scheduling Late MDU discarding
TSS [10]	Global	Unknown	Periodical and no periodical	Live	Inter and group	Distributed control scheme	–	No	Timestamps	Receiver	–	Event-based synchronization control Playout duration extension Rollback-based techniques

Please cite this article as: F. Boronat, et al., Multimedia group and inter-stream synchronization techniques: A comparative study, Informat. Systems (2008), doi:10.1016/j.is.2008.05.001

ILA [61]	Global	Unknown	Periodical and no periodical	Both	Group	Distributed control scheme	–	No	Timestamps	Receiver	–	Event-based synchronization control Preventive MDU/event discarding Reactive events discarding
[9]	Local	Unknown	Periodical and no periodical	Both	Intra and inter	–	–	No	Timestamps Seq. number	Receiver	–	Initial transmission and playout instants Reactive skips and pauses (silences for audio) Playout duration extensions or reductions Late MDU discarding
[24]	Local	Unknown	Periodical and no periodical	Stored (video and metadata)	Inter	–	M/S streams	No	Synchronization information in MDUs	Source and receiver	–	Periodical insertion of synchronized information in the video stream to achieve media synchronization of video and metadata Skips and pauses
ESMS [69]	Global	Known	Periodical and no periodical	Both	Intra and group (VTR for intra)	Synchronization maestro scheme	M/S receivers	Yes	Timestamps Seq. number	Source and Receiver	No	Skipping and VTR [76] techniques for intra-stream synchronization Same techniques than in SMS [14], but with two reference output timings and different methods
[28]	Global	Unknown	Periodical	Live	Intra and inter	–	M/S streams	No	Timestamps	Receiver	Yes	Preventive modification of the length of the silent periods for intra-stream synchronization Coarse synchronization at the beginning of a talkspurt Playout duration extensions or reductions M/S streams switching
ECSA [65]	Global	Unknown	Periodical and no periodical	Live	Inter	–	M/S streams	No	Event information Timestamps	Source and receiver	–	Event-based synchronization control Rollback-based techniques

ACME, abstractions for continuous media; ALPSMS, application-level protocol for synchronized multimedia sessions; ASP, application synchronization protocol; BS, bucket synchronization; DSC, distributed control scheme; ECSA, event correlation synchronization algorithm; FGP, feedback global protocol; FP, feedback protocol; FSP, flow synchronization protocol; ILA, interactivity-loss avoidance; LL-TW, local-lag and timewarp algorithms; LSA, lip-synchronization algorithm; MCP, multi-flow conversation protocol; MoSync, mobile synchronization algorithm; MSTP, multimedia synchronization transport protocol; PARK, paused-and-run k-stream multimedia synchronization control scheme; RTP-FGP, RTP-based feedback-global protocol; RVTR, retransmission with VTR; SMS, synchronization maestro scheme; SSP, stream synchronization protocol; TSS, trailing state synchronization; VTR, virtual time rendering algorithm.

environments, some techniques described for sources in this paper are used by local base stations in the cells where the mobile receiver is located (as in [32–34]).

4.2.1. Basic control techniques

Basic control, which consists of appending synchronization information (timestamp, sequence number, etc.) to MDUs and buffering the data at the receiver side, is essential for all algorithms. The following techniques have been found in all the solutions we have found.

(a) Source control
 The basic control techniques executed by the source can consist of introducing some *information useful for synchronization* in the headers of the MDUs, such as *timestamps*, *sequence numbers* (*identifiers*), *sequence marking* (streamlined time stamps), *event information and/or source identifiers* (see column '*Synchr. Information*' in Table 2). The use of *timestamps* is not needed when, for example, the generation of MDUs is periodical, and the use of only sequence numbers would be sufficient.
 Moreover, in some cases, the source can include temporal or event marks to force the inter-stream resynchronization at specific instants of time in the playout process (we call this process *Coarse* [38,57–59,76,88] or *Event* [10,11,35,53,61,65] *synchronization*). In the example presented in Section 3, the solution uses timestamps and sequence numbers, included in RTP/RTCP packets' headers.

(b) Receiver control
 Nearly all the solutions use *buffering techniques* at the receiver side. The reception buffers are used to keep MDUs until their playout instants arrive, according to certain synchronization information, and to smooth out the effects of the network jitter.

4.2.2. Preventive control techniques

Preventive control consists of techniques used to avoid asynchrony. We have found the following ones, according to their location.

(a) Source control
 For stored media content, source usually will be able to transmit MDUs according some synchronization information (for example, timestamps). This technique is used in most of the studied solutions. The source can draw up a schedule for transmission according to deadline times [25,27,32–34,57–59,89], but this technique is only valid for stored content transmission. We call this technique *deadline-based transmission scheduling*. If the source is able to know, for each MDU, its size, deadline transmission and the network delay limits (maximum and minimum delays or at least the probability distribution function of the delay), it will be able to schedule the transmission of MDUs according to those temporal requirements. In [32] both the server and local base station (wireless environment) schedule the transmission of MDUs. Boukerche et al. propose that the closest base station can schedule the packets for the playout of MDUs on the mobile receiver in several queuing policies (a FIFO [32], PQ, RR or WFQ [34] orders) according to the network conditions and applications' requirements.
 Another technique to prevent asynchrony consists of the *initial transmission and/or playout instant calculation*. As we explained above (example in Section 3), the source can prevent from an initial asynchrony situation at the starting of the playout of the different media streams. It can do it by calculating the *initial playout instant* of the presentation (common for all the receivers and streams) and communicating it to all the receivers before the transmission of media streams starts. So, all the receivers will start the playout at the same time. The initial playout instant calculation technique has been used in [2,8,9,12–14,26,29–31,57–59,62,89]. The initial transmission instant calculation technique has been used in [2,8,9,25,29,32–34,39,40,47,57–59,62].
 The source can also use a technique based on *interleaving MDUs* of different media streams in only one transport stream (as in [73,74]). This technique improves the inter-stream synchronization quality but may degrade the intra-stream synchronization quality in those streams sensitive to network jitter.

(b) Receiver control
 In some cases, the receiver's playout process will do *preventive skips of MDUs (eliminations or discardings) and/or preventive pauses of MDUs (repetitions or insertions)* depending on the playout buffer's size [16,61,70,71,73,74,76,78,89–92]. It is also possible to insert dummy (noise) data, instead of '*pausing*' (or stopping) the playout process. When using multi-level coding systems, such as MPEG, the receiver can discard some MDUs with low priority (e.g., B-frames in MPEG), according to the receiver buffer occupation. If the receiver can estimate the network delay experienced by the MDUs, it may *change the buffering waiting time of them* [76,2,62,66]. Some authors propose to enlarge or shorten the silence periods of the audio streams of the applications [28]. This technique is not valid for musical applications, where the silences are as important as the sounds.

4.2.3. Reactive control techniques

Reactive control is used to recover synchronization after asynchrony occurs. Approaches such as reactive skipping and pausing, shortening/extending playback duration and virtual local time contraction or expansion (referred as virtual local time control hereafter) all belong to reactive control. We have found the following reactive techniques:

(a) Source control
 On the one hand, the source can *adjust the transmission rate (timing)* changing the transmission period. If the source is capable of knowing the existing asynchrony between the streams' playout processes

[5,9,25,39,40,54–56,60,93,94], it will be able to change the transmission timing (for example, by changing the transmission period or by skipping or pausing MDUs). For example, in the PARK[3] approach [39,40], according to a TCP-like scheme, (a) the source could quickly decrease its transmission rate when network congestion is detected in order to release the congestion situation; (b) the source could slowly increase its transmission rate when the network congestion is released in order to utilize available bandwidth as much as possible.

On the other hand, if the source detects that recovery of synchrony is difficult for the receiver, it can *decrease the number of media streams transmitted* [70,71,73,74, 76,78,90–92,95]. For example, in audio and video synchronization case, if asynchrony is detected and this situation persists, the source could stop the transmission of the video stream temporarily, and, so, when it detects the receiver has recovered, the source could restart the transmission of the video stream.

When using multilevel coding systems, such as MPEG, the source can *drop low-priority MDUs* (for instance, B-frames in MPEG), as in [4,25,54–56,89], according to some QoS parameters, such as the network congestion or MDU loss rates.

(b) Receiver control

From the receiver point of view, it can take some actions for recovering from detected asynchrony situations. On the one hand, the most popular technique, due to its easy implementation, and used by the solution described in Section 3, consists of *reactive skips (eliminations or discardings) and/or reactive pauses (repetitions or insertions)*. This technique has been used by many authors. As an example, we can cite the case in which a receiver detects the playout point of the MDU it is processing has expired, because it has arrived too late. Then, it can choose between to playout it and discard consecutive MDUs already received (for example, [2,7,15,62,70,71,73–76, 78,90,93,95]), or to discard it directly (for example [9,12–14,19,30,31,35,39,40,54–56,66,96]). For audio streams, to deal with loses and delayed MDUs, a solution can be not to play anything [9]. In [66], the MDUs of several streams sent at the same instant (with the same timestamp) are considered as a synchronous group, so the playout of a MDU of a stream is *stopped* until the MDUs of other streams in the same group do not reach the receiver.

Experimental experience has demonstrated that abrupt skipping or pausing can result in playout gaps. Such gaps may dramatically degrade presentation effectiveness, especially when lip-synchronization is involved.

Some authors (for example, [57–59]) use the following nomenclature: blocking and restricted blocking policies. In a **Blocking Policy**, for slow streams, i.e. streams in which the MDUs do not arrive at the receiver in time to meet their respective playout deadlines, the playout process can be *blocked* or *suspended*, until the late MDU arrives. In a **restricted blocking policy**, the playout process blocks for a pre-specified period of time only, while it waits for the current MDU to arrive. After the waiting period with non-arrivals of late MDUs, it may playout the most recent stored MDUs, skipping the late MDUs altogether.

On the other hand, when receiver buffer starvation is detected, to avoid playout gaps, the receiver can opt to *playout repeatedly the last MDU* until the next one arrives [30,31,57–59] or to make *playout duration extensions or reductions* (*playout rate adjustments*) rather than abruptly skipping or pausing the presentation. To gradually recover from an asynchrony situation without degrading the playout quality (not noticed by application end users), the receiver can shorten or extend the playout duration of each MDU until the synchronization has been recovered [2–5,8–10,26,28–31,36,37,41,48–56,60,62,70,71,73–76,78, 88,90,93,94,97]. If the duration is shortened, it implies an acceleration in the playout (fast-forwarding), but without skips of MDUs, meanwhile an extension of the duration implies to slow down the playout (but without MDUs repetitions). This technique is also used to adapt the playout of a stream to the playout of another stream (inter-stream synchronization). In the case of voice streams, and in specific applications, only the playout of the MDUs of silent periods could be shortened or extended to affect as little as possible to the quality perceived by the users.

The extension of the playout duration technique is similar to the use of preventive pauses, because the latter can be made by enlarging the output or playout time of the MDUs. Nevertheless, the former is a reactive technique while the latter is preventive.

Some authors propose the *use of a virtual time with contractions or expansions* to get the desired synchronization. This way the MDUs are played using a virtual time axis different from the real-time axis. This technique has been used in many proposals [7,15,22,30,31,50–52,70,71, 73–76,78,89,90,93]. Virtual time expands or contracts according to the amount of delay jitter of the received MDUs.

In [3], *timestretching* the audio at the beginning of each utterance[4] is proposed for lip-synchronized videoconference systems. The audio stream can be time-stretched by re-sampling and interpolating the original audio packet.

For example, in Fig. 10, the playout of stream A and stream B becomes asynchronous at instant t_0 after MDU 30. The receiver recovers resynchronization control at point t_1 such that the presentation becomes synchronous when MDU 37 of both streams is being played out at the same time. The stream A playout process has played

[3] *Paused-and-run k-stream multimedia synchronization control scheme.*

[4] The beginning of an utterance is defined as the moment when the audio volume exceeds a silence threshold, the maximum measured audio volume when the user is not talking. The end of an utterance is defined as the moment when the audio volume is less than the silence threshold [3].

out duplicated MDUs and has adjusted the playout rate, whereas the stream B playout process has stopped its playout during a short interval of time (silence).

Normally, to obtain an inter-stream synchronization, a *master/slave scheme* is used, defining one of the streams as the *master stream*, and considering the others as the *slave streams* (as in the example described in Section 3 [2–5,7,12–14,24,26,28,30,31,35,38–41,48,50–52,60,62,63, 65,69–71,73–76,78–84,88,90,93,94,97–100]. There are some proposals in which the master and slave roles are exchanged dynamically, allowing master–slave switching [2,4,7,12–14,28,30,31,39,40,62,77,98]. This way, when the asynchrony of a slave stream exceeds a threshold value, the receiver can switch the roles, and consider that stream as the new master one, making the needed required adaptations. In [77], the master role is switched between streams according to their *global importance* in virtual environments.

Also in group synchronization, a master/slave schema can be used but regarding the receivers [15,26,30,31 ,43,78]. The master receiver playout timing is taken as the reference for updating the playout timings of the other receivers (slaves). As in the above case, in some algorithms the roles can be switched between receivers [26,30,31].

Fig. 11 shows an example of a master/slave inter-stream synchronization control. When the playout process of the slave stream ends its playout at a synchronization point and the playout process of the master stream has not finished yet, the playout process of the slave stream pauses, repeating the playout of the last MDU, or stopping its playout (blocking) until the master stream playout process finishes (Fig. 11a). When a slave stream playout process has not played some MDUs (because they have not arrived yet or because they arrived too late), and the master stream playout process has already finished its playout at a synchronization point, the slave stream playout process discards the late MDUs to maintain synchronization with the master stream playout process (Fig. 11b).

Ishibashi et al. [77] carried out subjective and objective assessment of the lip-synchronization quality of nine receiver-based synchronization control schemes, which consist of combinations of the following four reactive control techniques: skipping, discarding, shortening and extension of output duration, and virtual time contraction and expansion. In this paper, those nine schemes are used for inter-stream synchronization purposes, whereas the VTR algorithm [76] is used for intra-stream synchronization. They conclude that a scheme which uses the shortening and extension of output duration and the virtual time contraction and expansion together for voice and the shortening and extension of output duration for video produces the best quality of lip-synchronization. They also confirmed that the skipping and discarding control is not suited to voice.

On the other hand, in [10,11,35,53,61,65], the concept of **event-synchronization** is introduced (in [35], the case of a tele-robotic system is presented, but this kind of synchronization is usually used in networked game applications). It is similar to inter-stream synchronization, but it uses event-based action reference instead of the time. Therefore, event-synchronization control is coarser-grained than inter-stream synchronization [100]. In order to keep a consistent view of the state of the application, some mechanisms to guarantee a global ordering of events are necessary. This can either be done by avoiding disordering (by waiting for all possible events to arrive), or by having mechanisms to detect and correct disordering. To achieve this goal, **conservative and optimistic synchronization algorithms** have been devised. These algorithms usually consist of a collection of logical processes that communicate by exchanging timestamped messages or events. In conservative algorithms [11], receivers are not allowed to advance their virtual clocks until all other receivers have acknowledged that they have completed the computation for the current time period. It is impossible for inconsistencies to occur since no receiver performs calculations until it has the same exact information as everyone else. Unfortunately, with this scheme it is impossible to guarantee any relationship between virtual time (game time) and wall-clock time. Optimistic algorithms [10,35,53,61,65] execute events before they know for sure that no earlier events could arrive, and then repair inconsistencies when they are wrong. Algorithms of this type are far better suited to interactive situations (i.e. networked games). In contrast to conservative approaches that avoid violations of the local causality constraint, optimistic methods allow violations to occur, but are able to detect and recover from potential inconsistency. We can find several different reactive techniques: receivers can *discard late events* [11] or use *rollback techniques*, such as maintaining late events and using them to compensate for inconsistency at the receiving end (in the Timewarp algorithm [53] this can cause an extra overhead in terms of memory space and computation for inconsistency compensation). In [53] *Rollback based techniques* are also exploited to reestablish the consistency of the game state. Copies of the states are maintained after command executions and events received after their playout time are stored locally instead of being dropped and used to compensate for the inconsistency among receivers' views. Then, visual rendering of significant events can be delayed (to avoid inconsistencies if corrections occur). The problem, in this case, is that the use of these realignment techniques may further impact on the responsiveness of the system. Trailing State Synchronization (TSS [10]) is another optimistic synchronization algorithm, which uses *dynamically changing states as the source of rollbacks* as opposed to static snapshots, which is the fundamental difference between it and Timewarp [53]. It maintains several instances of the applications running with different synchronization delays. In [61], a proactive event discarding mechanism relying on the discrimination of obsolete events is used (obsolete events are discarded with a probability depending on the level of interactivity). In TSS inconsistencies are detected by detecting when the leading state and the correct state diverge, and are corrected at that point. In [65], state rollback is executed only when there exists an event whose timestamp is within the rollback time and the event is related with others that should be executed

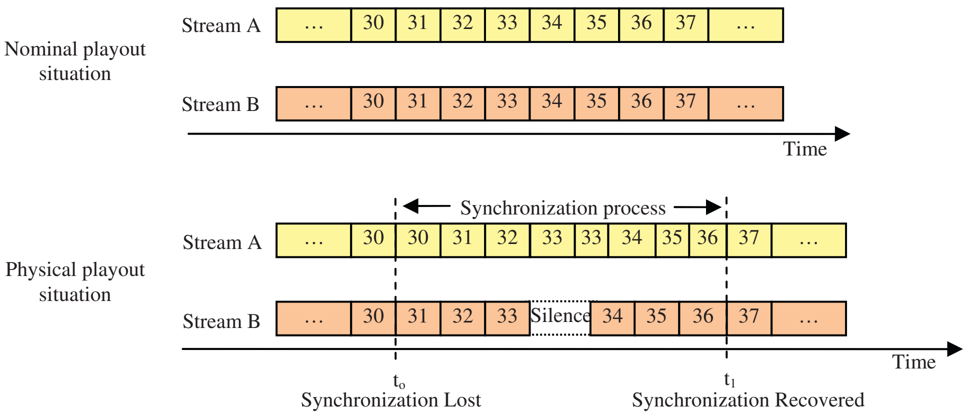


Fig. 10. Example of reactive control techniques at the receiver site.

after it but has been previously performed (correlated events).

4.2.4. Common control techniques

We have found several techniques that can be used as a means to prevent (preventive) or correct (reactive) asynchrony, from the source or the receiver side.

(a) Source control
Some authors [8,25,29,76,80] propose that the source *skips or pauses MDUs in the transmission process*, according to feedback information from the receivers (to prevent and/or correct the asynchrony situations). Moreover, the source could send empty MDUs, instead of skipping, when the generation rate is lower than the transmission rate [101].
When we consider stored contents [25], the source can *advance the transmission timing dynamically depending on the network delay estimation* (made by both the source and/or the receivers). For example, the timing can be advanced by skipping MDUs in the transmission.
In [93], the *adjustment of the input rate* is proposed. The source can vary the clock frequency of the input device, according to the obtained synchronization quality. In this work the *interpolation of data* is also proposed to adjust the effective input rate.
Another technique that can be used is the *Media Scaling*. Layered multicast is an example of it and more or less streams can be transmitted depending on the network conditions [74,102]. For example, the temporal or spatial resolution of the video stream can be changed depending on the network load.

(b) Receiver control
One of the techniques included in this Group is the *adjustment of the playout rate* by modifying the playout device's clock frequency, according to the obtained synchronization quality. In [2,62,93] the receivers adjust the playout rate according to the size of the playout buffer.
In [93], another technique is proposed: the *data interpolation* in the receiver side to adjust the effective output rate. Table 1 summarizes all the techniques described above.

5. Comparison

We have found numerous approaches to the modeling and execution of multimedia synchronization scenarios. Unfortunately, these approaches are difficult to compare and evaluate due to their varied theoretical bases and modeling techniques [23].

In Table 2 we summarize, chronologically ordered, all the identified synchronization solutions, presenting the above-described techniques, and other factors of interest, such as the following ones:

- *Clocks*: Table indicates if the clock signal used by the algorithm is globally synchronized (global reference) or if it is available only locally (local reference).
- *Network delay limits*: The need for the solution to know in advance these limits or their probability distribution function is indicated.
- *MDU generation periodicity*: The solution can have been developed to work with transmission of streams in which the generation of MDUs is periodical or not.
- *Stored or live contents*: Some solutions have been developed for transmission of stored content, live content or for both content types.
- *Synchronization type included* in the solution (intra-stream, inter-stream and/or group). In this case, as mentioned before, only those solutions which include inter-stream or group Synchronization have been classified. Solutions for group synchronization were not included in the classification presented in [22]. Those which only include intra-stream synchronization have not been considered.
- *Master/slave relationship*: The existence of master/slave relationships (between streams or receiver) is indicated in the table.
- *Group synchronization techniques*: In the solutions which include group synchronization techniques, the technique/s included are also indicated (*master/slave receiver scheme*, *SMS* and/or *distributed control scheme*).

Please cite this article as: F. Boronat, et al., Multimedia group and inter-stream synchronization techniques: A comparative study, Informat. Systems (2008), doi:10.1016/j.is.2008.05.001

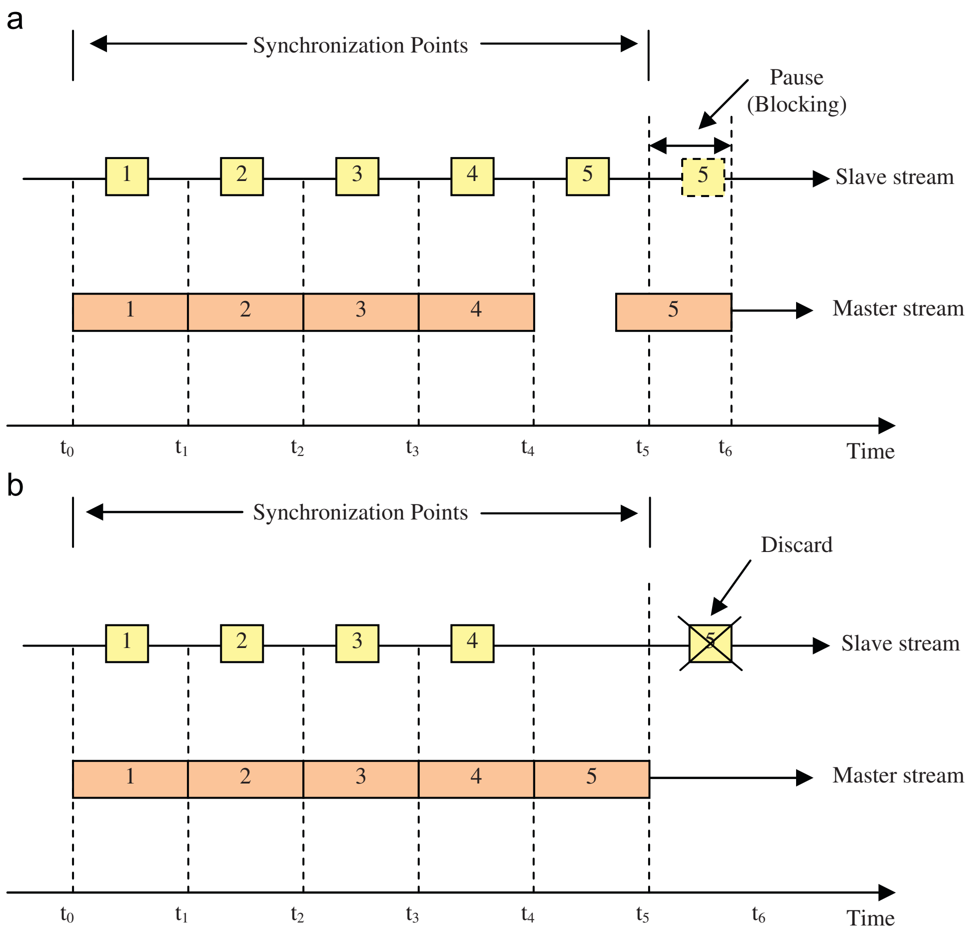


Fig. 11. Example of master/slave inter-media synchronization techniques. (a) Pausing MDUs playout or stopping MDUs playout technique; (b) discarding MDUs technique.

- *Feedback utilization*: Some solutions use feedback information included in the messages sent back from the receivers to the sources. The use of feedback techniques for synchronization purposes is indicated.
- *Synchronization information*: The information useful for synchronization included in the transmitted MDUs (if there is any) is indicated.
- *Location of the synchronization techniques*: The location of the synchronization techniques is important. The synchronization control can be done by the source/s or by the receiver/s.
- *RTP use*: New proposals use or allow the use of RTP/RTCP protocol [85]. Older proposals do not use it.
- *Synchronization techniques*: The most representative techniques included in each solution have been indicated in the table.

In the first column (*Name*) we present the name the authors assigned to their proposal and its main references. If there is no name, we only put the references.

The gaps in the table indicate that the factor related to the column is not considered or included by the solution or, possibly, we have not found any mention of the use or the inclusion of that factor in the references in which the solution is described.

We can see that there are a large variety of synchronization solutions, and they differ in terms of goals and application scenarios. They have been defined for many different conditions and environments; therefore, in each one a different combination of several synchronization techniques is used. It has not been an easy job to try to find the relationships between them and to make the qualitative comparison. Perez and Little [23] explain why there are so many synchronization frameworks, how a multimedia scenario can be represented with different temporal specification schemes, and why some specification schemes cannot model all scenarios.

The solution proposed by the authors [30,31], described in Section 3 has been emphasized (in bold type).

6. Conclusions

In this paper, a comprehensive qualitative comparison between the 53 most relevant multimedia synchronization solutions has been described. As far as the authors know this is the largest one that has been published. It has been done taking into account some critical issues in the multimedia synchronization field. Once the three types of multimedia synchronization (intra-stream, inter-stream and group) and the main synchronization techniques have been described, only those solutions that provide inter-stream and group synchronization have been considered in our survey. They have been studied and we have specified which techniques are included in each solution. All this information has been tabulated, with the solutions chronologically ordered. For a better understanding of the paper we have included our solution as an example of multimedia group and inter-stream synchronization proposal, which includes some of the described techniques and uses modified standard protocols, such as RTP/RTCP.

Although some of the references at the end of the paper may appear dated, the authors have chosen to use the references of the papers in which the solutions were described for the first time. Many of the solutions have been used and tested subsequently in new environments by their authors and those papers have also been referenced. As an example, among others, we have found the VTR algorithm [76] being used (slightly enhanced in some cases) recently in media synchronization between voice and movement of avatars in networked virtual environments [70], for group synchronization control for haptic media in networked virtual environments [44,45], in media games [48,69], in collaborative work scenarios [45,69,71], for media synchronization in wireless networks [46,87], and in a remote haptic drawing system [72]. Despite all these papers, we have maintained the VTR's original reference of 1995 [76].

Our main aim has not been to classify the solutions from best to worst because, as discussed before, they all differ in terms of goals and application scenarios (each one has been developed and is suitable for specific scenarios and application conditions). For this reason it is difficult (if not impossible) to compare all the solutions quantitatively. Moreover, in the references there are papers in which we can find the evaluation in the specific scenarios (some solutions have been evaluated and compared with other solutions in the related papers—hyphenated along the text of the paper). It would be very complicated (if not impossible) to implement and evaluate all them in the same scenario in order to compare the results about efficiency or synchronization QoS. In many cases we have only found a quite short paper describing the solution. For this reason we have chosen some objective issues that allow us to compare them, qualitatively at least.

In Table 2, the solutions have been ordered chronologically. A very important issue to emphasize is the fact that most of the modern solutions use feedback and time information (timestamps) included in the RTP/RTCP protocols, as the authors' proposal, described in Section 3, does. Until RTP was chosen as an standard for the time-dependent multimedia streams transmission protocol, most of the solutions did not follow nor use any standard protocol but defined new synchronization protocols with new data packet formats including time information and new control messages for feedback. Using RTP, the solutions take advantage of the use of control RTCP report packets for including feedback information or useful information for multimedia synchronization purposes. Modern solutions, as the ones described in [30,31,49], use RTP/RTCP and, moreover, some old solutions have been implemented subsequently using RTP (for example, VTR in [74,75]).

Our main aim is that this comparison should be very useful to new researchers in multimedia fields to understand quickly the most common synchronization techniques and which ones are used by each solution. Novel researchers have a very valuable starting point to choose and study the solutions they are interested in and to develop new solutions using the described techniques they choose.

Acknowledgments

Authors want to thank Denis Shasha, the Editor-in-Chief of the journal, and the reviewers for their comments that helped us improve the quality of this paper substantially.

References

[1] G. Blakowski, R. Steinmetz, A media synchronization survey: reference model, specification and case studies, IEEE J. Sel. Areas Commun. 14 (1) (1996) 5–35.
[2] K. Rothermel, T. Helbig, An adaptive stream synchronization protocol, in: Proceedings of the Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, New Hampshire, USA, April 1995, pp. 189–202.
[3] M. Chen, Low-latency lip-synchronized videoconferencing system, Proceedings of the Conference on Human Factors in Computing Systems (CHI2003), Fort Lauderdale, FL, USA 5 (1) (2003) 465–471.
[4] I. Kouvelas, V. Hardman, A. Watson, Lip synchronization for use over the internet: analysis and implementation, in: Proceedings of the IEEE Conference on Global Communications, GLOBECOM'96, vol. 2, London, UK, November 1996, pp. 893–898.
[5] L. Qiao, K. Nahrstedt, Lip synchronization within an adaptive VoD, SPIE Multimedia Computing and Networking 1997, San José, CA, USA, February 1997, pp. 170–181.
[6] C. Liu, Y. Xie, M.J. Lee, T.N. Saadawi, Multipoint multimedia teleconference system with adaptative synchronization, IEEE J. Sel. Areas Commun. 14 (7) (1996) 1422–1435.
[7] Y. Xie, C. Liu, M.J. Lee, T.N. Saadawi, Adaptive multimedia synchronization in a teleconference system, Multimedia Syst. 7 (4) (1999) 326–337.
[8] E. Biersack, W. Geyer, Synchronized delivery and playout of distributed stored multimedia streams, Multimedia Syst. 7 (1) (1999) 70–90.
[9] S.S. Manvi, P. Venkataram, An agent based synchronization scheme for multimedia applications, J. Syst. Software (JSS) 79 (5) (2006) 701–713.
[10] E. Cronin, B. Filstrup, S. Jamin, A.R. Kurc, An efficient synchronization mechanism for mirrored game architectures, Multimedia Tools Appl. 23 (l) (2004) 7–30.
[11] C. Diot, L. Gautier, A distributed architecture for multiplayer interactive applications on the Internet, IEEE Network 13 (4) (1999) 6–15.
[12] C.M. Huang, C. Wang, J.M. Hsu, Formal modeling and design of multimedia synchronization for interactive multimedia presentations in distributed environments, in: International Conference on Consumer Electronics 1998, ICCE 1998, Digest of Technical Papers, June 1998, pp. 458–459.
[13] C.M. Huang, C. Wang, Synchronization for interactive multimedia presentations, IEEE Multimedia 5 (4) (1998) 44–62.

[14] C.M. Huang, C. Wang, C.H. Lin, Interactive multimedia synchronization in the distributed environment using the formal approach, IEE Proc. Soft. 147 (4) (2000) 131–146.
[15] Y. Ishibashi, S. Tasaka, H. Miyamoto, Joint synchronization between stored media with interactive control and live media in multicast communications, IEICE Trans. Commun. E85-B (4) (2002) 812–822.
[16] D. Köhler, H. Müller, Multimedia playout synchronization using buffer level control, in: Proceedings of the Second International Workshop on Advanced Teleservices and High-Speed Communication Architectures, LNCS, vol. 868, Heidelberg, Germany, September 1994, pp. 167–180.
[17] N. Laoutaris, I. Stavrakakis, Intrastream synchronization for continuous media streams: a survey of playout schedulers, IEEE Network Mag. 16 (3) (2002) 30–40.
[18] Y.H. Wang, H.Z. Lin, Cooperating intelligent mobile agents mechanism for distributed multimedia synchronization, in: Proceedings of the International Conference on Multimedia and Expo (ICME'00), vol. 2, New York, USA, July/August 2000, pp. 743–746.
[19] A. Furfaro, L. Nigro, F. Pupo, Multimedia synchronization based on aspect oriented programming, Microprocessors Microsystems 28 (2) (2004) 47–56.
[20] F. Boronat, J.C. Guerri, Analysis and comparison of multimedia inter-stream and group synchronization algorithms, IEEE Latin Am. Trans. 3 (5) (2005) 23–32 (in Spanish).
[21] L. Ehley, B. Furht, M. Ilyas, Evaluation of multimedia synchronization techniques, in: Proceedings of the International Conference Multimedia Computing and Systems, ICMCS 94, Boston, MA, USA, May 1994, pp. 514–519.
[22] Y. Ishibashi, S. Tasaka, A comparative survey of synchronization algorithms for continuous media in network environments, in: Proceedings of the 25th IEEE Conference on Local Computer Networks, Tampa, FL, USA, November 2000, pp. 337–348.
[23] M.J. Perez-Luque, T.D.C. Little, A temporal reference framework for multimedia synchronization, IEEE J. Sel. Areas Commun. 14 (1) (1996) 36–51.
[24] H. Abe, H. Shigeno, K. Okada, Media synchronization method for video hypermedia application based on extended event model, in: Proceedings of IEEE International Conference on Multimedia and Expo, Toronto, Canada, July 2006, pp. 1289–1292.
[25] Z. Ali, A. Ghafoor, C.S. George Lee, Media synchronization in multimedia web using a neuro-fuzzy framework, IEEE J. Sel. Areas Commun. 18 (2) (2000) 168–183.
[26] I.F. Akyildiz, W. Yen, Multimedia group synchronization protocols for integrated services networks, IEEE J. Sel. Areas Commun. 14 (1) (1996) 162–173.
[27] S. Baqai, M.F. Khan, M. Woo, S. Shinkai, A.A. Khokar, A. Ghafoor, Quality-based evaluation of multimedia synchronization protocols for distributed multimedia information systems, IEEE J. Sel. Areas Commun. 14 (7) (1996) 1388–1403.
[28] I. Bartoli, G. Iacovoni, F. Ubaldi, A synchronization control scheme for videoconferencing services, J. Multimedia 1 (4) (2007) 1–9.
[29] E. Biersack, W. Geyer, C. Bernhardt, Intra- and inter-stream synchronization for stored multimedia streams, in: Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Hiroshima, Japan, June 1996, pp. 372–381.
[30] F. Boronat, J.C. Guerri, M. Esteve, J.M. Murillo, RTP-based feedback global protocol integration in Mbone tools, EUROMEDIA 2002, Modena, Italy (Best Paper Award) April 2002.
[31] F. Boronat, Specification and evaluation of a Multimedia group synchronization algorithm, Ph.D. Thesis, Polytechnic University of Valencia (UPV), Spain, April 2004, 410pp (in Spanish).
[32] A. Boukerche, S. Hong, T. Jacob, An efficient synchronization scheme of multimedia streams in wireless and mobile systems, IEEE Trans. Parallel Distributed Syst. 13 (9) (2002) 911–923.
[33] A. Boukerche, S. Hong, T. Jacob, Synchronization and handoff management schemes for wireless multimedia systems, Comput. Networks: Int. J. Comput. Telecommun. Networking 41 (3) (2003) 347–362.
[34] A. Boukerche, H. Owens, Media synchronization and QoS packet scheduling algorithms for wireless systems, Mobile Networks Appl. 10 (1–2) (2005) 233–249.
[35] I. Elhajj, N. Xi, B. Song, M.-M. Yu, W.T. Lo, Y.H. Liu, Transparency and synchronization in supermedia enhanced Internet-based teleoperation, in: Proceedings of the 2002 IEEE International Conference on Robotics & Automation, Washington, USA, May 2002, pp. 2713–2718.
[36] J. Escobar, C. Partridge, D. Deutsch, Flow synchronization protocol, IEEE/ACM Trans. Networking 2 (2) (1994) 111–121.
[37] A.J. González, H. Abdel-Wahab, Light-weight stream synchronization framework for multimedia collaborative applications, in: Proceedings of the Fifth IEEE Symposium on Computers and Communications, Antibes-Juan Les Pins, France, July 2000, pp. 398–403.
[38] J.C. Guerri, M. Esteve, C.E. Palau, V. Casares, Feedback flow control with hysteresial techniques for multimedia retrievals, Multimedia Tools Appl. 13 (3) (2001) 307–332.
[39] C.M. Huang, H.Y. Kung, J.L. Yang, PARK: a paused-and-run k-stream multimedia synchronization control scheme, in: Proceedings of the 20th International Conference on Distributed Computing Systems, Taipei, Taiwan, April 2000, pp. 272–279.
[40] C.M. Huang, H.Y. Kung, J.L. Yang, Synchronization and flow adaptation schemes for reliable multiple-stream transmission in multimedia presentations, J. Syst. Software 56 (2) (2001) 133–151.
[41] Y. Ishibashi, S. Tasaka, A group synchronization mechanism for stored media and its measured performance in a heterogeneous network, IEICE Trans. Commun. E82-B (7) (1999) 1009–1018.
[42] Y. Ishibashi, S. Tasaka, A group synchronization mechanism for live media in multicast communications, in: Global Telecommunications Conference, 1997 (IEEE GLOBECOM' 97), November 1997, pp. 746–752.
[43] Y. Ishibashi, S. Tasaka, Y. Tachibana, Adaptive causality and media synchronization control for networked multimedia applications, in: Proceedings of the IEEE International Conference on Communications, vol. 3, Helsinki, Finland, June 2001, pp. 952–958.
[44] Y. Ishibashi, T. Hasegawa, S. Tasaka, Group synchronization control for haptic media in networked virtual environments, in: Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Chicago, USA, March 2004, pp. 106–113.
[45] H. Kaneoka, Y. Ishibashi, Effects of group synchronization control over haptic media in collaborative work, in: Proceedings of the 14th International Conference on Artificial Reality and Telexistence (ICAT'04), Coex, Korea, November/December 2004, pp. 138–145.
[46] S. Tasaka, Y. Ishibashi, M. Hayashi, Inter-destination synchronization quality in an integrated wired and wireless network with handover, in: Global Telecommunications Conference, GLOBECOM'02, vol. 2, Taipei, Taiwan, November 2002, pp. 1560–1565.
[47] Y. Ishibashi, S. Tasaka, A distributed control scheme for group synchronization in multicast communications, in: Proceedings of the International Symposium Communications, Kaohsiung, Taiwan, November 1999, pp. 317–323.
[48] Y. Ishibashi, S. Tasaka, A distributed control scheme for causality and media synchronization in networked multimedia games, in: Proceedings of the 11th International Conference on Computer Communications and Networks, Miami, USA, October 2002, pp. 144–149.
[49] C.C. Kuo, M.S. Chen, J.C. Chen, An Adaptive transmission scheme for audio and video synchronization based on real-time transport protocol, in: Proceedings of the IEEE International Conference on Multimedia and Expo, August 2001.
[50] H. Liu, M. El Zarki, A synchronization control scheme for real-time streaming multimedia applications, in: Proceedings of the 13th Packet Video Workshop, Nantes, France, April 2003.
[51] H. Liu, M. El Zarki, On the adaptive delay and synchronization control of video conferencing over the internet, in: Proceedings of the International Conference on Networking 2004 (ICN'04), February 2004.
[52] H. Liu, M. El Zarki, Adaptive delay and synchronization control for Wi-Fi based AV conferencing, in: Proceedings of the First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, Dallas, TX, USA, October 2004, pp. 84–91.
[53] M. Mauve, J. Vogel, V. Hilt, W. Effelsberg, Local-lag and timewarp: providing consistency for replicated continuous applications, IEEE Trans. Multimedia 6 (l) (2004) 47–57.
[54] M. Mielke, A. Zhang, A multi-level buffering and feedback scheme for distributed multimedia presentation systems, in: Proceedings of the Seventh International Conference Computer Comm. and Networks, Lafayette, LA, USA, October 1998, pp. 219–226.
[55] T. Vinod, A. Zhang, Dynamic playout scheduling algorithms for continuous multimedia streams, Multimedia Syst. 7 (4) (1999) 312–325.
[56] A. Zhang, Y. Song, M. Mielke, NetMedia: streaming multimedia presentations in distributed environments, IEEE Multimedia 9 (1) (2002) 56–73.

Please cite this article as: F. Boronat, et al., Multimedia group and inter-stream synchronization techniques: A comparative study, Informat. Systems (2008), doi:10.1016/j.is.2008.05.001

[57] R. Mussauer, F. Paiva, Gonçalves, P.A. da S. Gonçalves, O.C. Muniz, SAMM: an integrated environment to support multimedia synchronization of pre-orchestrated data, in: IEEE International Conference on Multimedia Computing and Systems 1999, vol. 2, Florence, Italy, 06/07/1999–06/11/1999, pp. 929–933.
[58] N.U. Qazi, M. Woo, A. Ghafoor, A synchronization and communication model for distributed multimedia objects, in: Proceedings of the ACM Multimedia'93, Anaheim, CA, August 1993, pp. 147–56.
[59] M. Woo, N.U. Qazi, A. Ghafoor, A synchronization framework for communication of pre-orchestrated multimedia information, IEEE Network 8 (1) (1994) 52–61.
[60] K. Nahrstedt, L. Qiao, Stability and Adaptation Control for Lip Synchronization Skews, Technical Report, Department of Computer Science, University of Illinois at Urbana-Champaign, May 1998.
[61] C.E. Palazzi, S. Ferretti, S. Cacciaguerra, M. Roccetti, On maintaining interactivity in event delivery synchronization for mirrored game architectures, in: IEEE Global Telecommunications Conference Workshops, Dallas, TX, USA, November/December 2004, pp. 157–165.
[62] K. Rothermel, T. Helbig, An adaptive protocol for synchronizing media streams, ACM/Springer Multimedia Syst. 5 (5) (1997) 324–336.
[63] N. Shivakumar, C. Sreenan, B. Narendran, P. Agrawal, The concord algorithm for synchronization of networked multimedia streams, in: Proceedings of the Second IEEE International Conference on Multimedia Computing and Systems, Washington, DC, USA, May 1995, pp. 31–40.
[64] S.H. Son, N. Agarwal, A model for specification and synchronization of data for distributed multimedia applications, Multimedia Tools Appl. 3 (2) (1992) 79–104.
[65] S. Xiang-bin, L. Fang, D. Ling, Z. Xing-hai, X. Yuan-sheng, An event correlation synchronization algorithm for MMOG, in: Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, vol. 1, Qingdao, China, July/August 2007, pp. 746–751.
[66] C.C. Yang, Design of the application-level protocol for synchronized multimedia sessions, in: Proceedings of the IEEE International Conference on Communications, 2002 (ICC'02), New York, USA, April/May 2002, pp. 2518–2522.
[67] R. Yavatkar, MCP: A protocol for coordination and temporal synchronization in collaborative applications, in: Proceedings of the 12th International Conference Distributed Computing Systems, Yokohama, Japan, June 1992, pp. 606–613.
[68] R. Yavatkar, K. Lakshman, Communication support for distributed collaborative applications, Multimedia Syst. 2 (2) (1994) 74–88.
[69] T. Hashimoto, Y. Ishibashi, Group synchronization control over haptic media in a networked real-time game with collaborative work, in: Proceedings of the Fifth ACM SIGCOMM workshop on Network and System Support for Games, Singapore, October 2006.
[70] Y. Ishibashi, I. Inazumi, S. Tasaka, T. Asano, Media synchronization between voice and movement of avatars in networked virtual environments, in: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Singapore, June 2004, pp. 134–139.
[71] Y. Ishibashi, T. Kanbara, S. Tasaka, Inter-stream synchronization between haptic media and voice in collaborative virtual environments, in: Proceedings of the 12th annual ACM international conference on Multimedia, New York, USA, October 2004, pp. 604–611.
[72] Y. Kurokawa, Y. Ishibashi, T. Asano, Group synchronization control in a remote haptic drawing system, in: Proceedings of the IEEE International Conference on Multimedia and Expo, Beijing, China, July 2007, pp. 572–575. Utiliza la rec ITU-R Rec. BT. 500 para evaluación subjetiva
[73] S. Tasaka, Y. Ishibashi, Media synchronization in heterogeneous networks: stored media case, IEICE Trans. Commun. E81-B (8) (1998) 1624–1636.
[74] S. Tasaka, Y. Ishibashi, A Performance Comparison of Single-Stream and Multi-Stream Approaches to Live Media Synchronization E81-B (11) (1998) 1988–1997.
[75] S. Tasaka, T. Nunome, Y. Ishibashi, Live media synchronization quality of a retransmission-based error recovery scheme, in: Proceedings of the IEEE International Conference on Communications, New Orleans, LA, USA, June 2000, pp. 1535–1541.
[76] Y. Ishibashi, S. Tasaka, A synchronization mechanism for continuous media in multimedia communications, in: Proceedings of the IEEE INFOCOM'95, Boston, MA, USA, April 1995, pp. 1010–1019.
[77] Y. Ishibashi, K. Tomaru, S. Tasaka, K. Inazumi, Group synchronization in networked virtual environments, in: Proceedings of the 38th IEEE International Conference on Communications 2003, vol. 2, Alaska, USA, May 2003, pp. 885–890.
[78] Y. Ishibashi, A. Tsuji, S. Tasaka, A group synchronization mechanism for stored media in multicast communications, in: Proceedings of the Sixth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, Kobe, Japan, April 1997, pp. 692–700.
[79] S. Ramanathan, P.V. Rangan, Continuous media synchronization in distributed multimedia systems, in: Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video, La Jolla, CA, USA, November 1992, pp. 289–296.
[80] S. Ramanathan, P.V. Rangan, Adaptive feedback techniques for synchronized multimedia retrieval over integrated networks, IEEE/ACM Trans. Networking 1 (2) (1993) 246–259.
[81] S. Ramanathan, P.V. Rangan, Feedback techniques for intra-media continuity and inter-media synchronization in distributed multimedia systems, Comput. J. 36 (1) (1993) 19–31.
[82] P.V. Rangan, S. Ramanathan, Designing an on-demand multimedia service, IEEE Commun. Mag. 30 (7) (1992) 56–64.
[83] P.V. Rangan, S. Ramanathan, T. Kaeppner, Performance of inter-media synchronization in distributed, heterogeneous multimedia systems, Comput. Networks ISDN Syst. 27 (4) (1995) 549–565.
[84] P.V. Rangan, S. Ramanathan, S. Sampathkumar, Feedback techniques for continuity and synchronization in multimedia information retrieval, ACM Trans. Inf. Syst. 13 (2) (1995) 145–176.
[85] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, RFC-3550, July 2003.
[86] D. Kutscher, J. Ott, The Message bus. a communication-integration infrastructure for component-based systems, White Paper, January 2000. ⟨http://www.mbus.org/mbuswp.pdf⟩.
[87] T. Nunome, S. Tasaka, Inter-destination synchronization quality in a multicast mobile ad hoc network, in: Proceedings of the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, Berlin, Germany, September 2005, pp. 1366–1370.
[88] M.C. Yuang, B.C. Lo, Y.G. Chen, P.L. Tien, A synchronization paradigm with QoS guarantees for multimediacommunications, in: Global Telecommunications Conference, GLOBECOM '99, vol. 1, Rio de Janeiro, Brazil, 12/05/1999–12/09/1999, pp. 214–220.
[89] L. Lamont, L. Li, R. Brimont, D. Georganas, Synchronization of multimedia data for a multimedia news on demand application, IEEE J. Sel. Areas Commun. 14 (1) (1996) 264–278.
[90] Y. Ishibashi, S. Tasaka, Y. Tachibana, Media synchronization and causality control for distributed multimedia applications, IEICE Trans. Commun. E84-B (3) (2001) 667–677.
[91] T.D.C. Little, F. Kao, An intermedia skew control system for multimedia data presentation, in: Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video, La Jolla, CA, USA, November 1992, pp. 121–132.
[92] T.D.C. Little, A framework for synchronous delivery of time-dependent multimedia data, Multimedia Syst. 1 (2) (1993) 87–94.
[93] D.P. Anderson, G. Homsy, A continuous media I/O server and its synchronization mechanism, IEEE Comput. 24 (10) (1991) 51–57.
[94] M. Correia, P. Pinto, Low-level multimedia synchronization algorithms on broadband networks, in: International Multimedia Conference, Proceedings of the Third ACM international conference on Multimedia, San Francisco, CA, USA, November 1995, pp. 423–434.
[95] K. Ravindran, V. Bansal, Delay compensation protocols for synchronization of multimedia data streams, IEEE Trans. Knowl. Data Eng. 5 (4) (1993) 574–589.
[96] H.Y. Chen, J.L. Wu, MultiSync: a synchronization model for multimedia system, IEEE J. Sel. Areas Commun. 14 (1) (1996) 238–248.
[97] S. Yoo, Realization of the synchronization controller for multimedia applications, in: Proceedings of the IEEE GLOBECOM'98, vol. 2, Sydney, Australia, December 1998, pp. 798–803.

Please cite this article as: F. Boronat, et al., Multimedia group and inter-stream synchronization techniques: A comparative study, Informat. Systems (2008), doi:10.1016/j.is.2008.05.001

[98] C.M. Huang, R.Y. Lee, Achieving multimedia synchronization between live video and live audio streams using QoS controls, Comput. Commun. 19 (5) (1996) 456–467.
[99] P.V. Rangan, S. Srihari, S. Rajan, Continuity and synchronization in MPEG, IEEE J. Sel. Areas Commun. 14 (1) (1996) 52–60.
[100] C.C. Yang, J.H. Huang, A Multimedia synchronization model and its implementation in transport protocols, IEEE J. Sel. Areas Commun. 14 (1) (1996) 212–225.
[101] P.N. Zarros, M.J. Lee, T.N. Saadawi, Interparticipant synchronization in real-time multimedia conferencing using feedback, IEEE/ACM Trans. Networking 4 (2) (1996) 173–180.
[102] S. McCanne, Van Jacobson, M. Vetterli, Receiver-driven layered multicast, in: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM SIGCOMM, Palo Alto, CA, USA, August 1996, pp. 117–130.

Lloret J, Garcia M, Tomás J *et al.* GBP-WAHSN: A group-based protocol for large wireless ad hoc and sensor networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(3): 461–480 May 2008

GBP-WAHSN: A Group-Based Protocol for Large Wireless Ad Hoc and Sensor Networks

Jaime Lloret, Miguel Garcia, Jesus Tomás, and Fernando Boronat

Department of Communications, Polytechnic University of Valencia, Camino Vera s/n, 46022, Valencia, Spain

E-mail: {jlloret, jtomas, fboronat}@dcom.upv.es; migarpi@posgrado.upv.es

Revised March 11, 2008.

Abstract Grouping nodes gives better performance to the whole network by diminishing the average network delay and avoiding unnecessary message forwarding and additional overhead. Many routing protocols for ad-hoc and sensor networks have been designed but none of them are based on groups. In this paper, we will start defining group-based topologies, and then we will show how some wireless ad hoc sensor networks (WAHSN) routing protocols perform when the nodes are arranged in groups. In our proposal connections between groups are established as a function of the proximity of the nodes and the neighbor's available capacity (based on the node's energy). We describe the architecture proposal, the messages that are needed for the proper operation and its mathematical description. We have also simulated how much time is needed to propagate information between groups. Finally, we will show a comparison with other architectures.

Keywords group-based protocol, group-based architecture, group-based routing algorithm, large networks

1 Introduction

Wireless ad hoc networks (WAHN) are simple networks in which a coordinator is not needed and the numbers of nodes and network topology are not predetermined. A wireless sensor networks (WSN) is a type of WAHN composed of nodes with sensing capability. There are several differences between WSN and WAHN[1]. WSNs usually have a larger number of nodes and are deployed in close proximity to the phenomena under study; the nodes mainly use a broadcast communication paradigm and the network topology can change constantly due, for example, to the fact that the nodes are prone to fail (they have limited power, computational capabilities and memory). Mobile wireless sensor networks (MWSNs) are WSNs with mobile sensors which are randomly deployed in an interesting area for sensing some phenomena. These mobile sensors collaborate with each other to form a sensor network with the capability of reporting sensed phenomena to a data collection point called sink or base station.

A mobile ad hoc network (MANET[2]) is a self-configuring network of mobile nodes connected by wireless technology. This type of network has an arbitrary topology. The network's wireless topology may change rapidly and unpredictably. Independently of the medium access method used[3], in recent years have many routing protocols been developed for these networks[4,5]. The nodes' mobility, the lack of stability of the topology, the lack of a pre-established organization and performing of the wireless communications are the reasons for not using the routing protocols developed for fixed networks.

Depending on the type of the information exchanged by the nodes and on the frequency by which they do it, the routing protocols in ad hoc networks are divided into three types: *proactives*, *reactives* and *hybrids*. The proactive protocols update the routing tables of all the nodes periodically, even though no information is being exchanged. When a topology change occurs, the routing table is updated and the routing protocol finds the best route to forward the information. A periodical control protocol message exchange allows this, but consumes bandwidth and energy. The reactive protocols only maintain routing routes in their tables when a node has to communicate with another node in the network. With these protocols, when a communication starts, as the right route is unknown, a route discovering message is sent. When the response is received, the route is included in the routing tables and the communication is established. The main disadvantage of these protocols is the latency at the beginning of the communications (route discovery time) but they improve the consumption of network and energy resources. Finally, hybrid protocols are a combination of the above two types, taking their advantages. These protocols divide

Regular Paper

ad hoc networks into different zones; consequently near nodes use proactive routing while far nodes use reactive routing.

The aforementioned networks and protocols do not have a predetermined topology, so they could be applied over different types of architectures such as Grids, cluster-based networks, group-based networks and so on.

A key problem in the planning of any kind of network is to design the communication topology. It means deciding how the peers are connected as well as how their messages are exchanged. Topologies can be characterized by several parameters such as the number of nodes in the network, the number of links or connections (hereafter both terms will be used without distinction in this paper) in the network and their bandwidth, the degree of the nodes and the diameter of the topology. On the other hand, communication topology design needs to address several conflicting requirements like, on the one hand, minimizing the overall network diameter, minimizing the convergence time, the infrastructure cost (total number of links), the book-keeping costs (the number of links maintained by each node) and the management cost, and, on the other hand, maximizing load distribution, reliability, efficiency, fault tolerance, the performance of the system, the scalability, and so on. Usually, optimizing on any requirements would be at the cost of others. Designing the optimal topology for a given set of constraints is a difficult problem. Over the years, topology design has received significant interest in many areas. In order to provide real-time infrastructures, reliable, available and efficient networks and QoS-aware distribution services, a topology-aware network is necessary[6,7].

While the physical topology defines how the nodes on a network are physically connected and the physical layout of the devices on the network, the logical topology defines how the nodes on the network communicate (i.e., the way the data passes through the network, with no regard for the physical interconnection between the devices). However, if the logical network is constructed randomly, nearby hosts in the logical network may be far away in the physical network. This may waste too many network resources, and hence degrade data delivery performance significantly.

In this paper, we present a proposal which uses a group-based topology and protocol over WAHSNs in order to improve their performance.

The remainder of the paper is organized as follows. Section 2 describes group-based architectures. Some application environments are presented in Section 3. Section 4 demonstrates that group-based topologies can improve some routing protocols such as Dynamic Source Routing Protocol (DSR), Optimized Link State Routing Protocol (OLSR) and ad hoc on demand distance vector (AODV) routing. The architecture operation and its analytical model are shown in Section 5. Protocol operation is shown in Section 6. Section 7 shows simulations to test our protocol. In Section 8, we compare our proposal with other types of networks. Finally, Section 9 summarizes the results and exposes future research.

2 Group-Based Topologies

The network topology defines how the nodes on a network are physically or logically connected (i.e., the physical layout of the devices on the network). Three types of network topologies can be distinguished:

1) Centralized Networks. In these topologies there could be no direct connection between nodes, and all nodes' messages could be mediated by a mediator, generally known as a central node. This single node acts as a gateway for all the nodes. These topologies have been used for many types of networks[8].

2) Decentralized Networks. Each node is able to connect directly with all other nodes, and messages are sent without intermediation via a central node. All nodes have the same responsibility and functionality in the network. No element in the network is essential for the system operation. A node in a decentralized topology can play three roles: *server*, *client* and *router*. Many types of networks have decentralized topologies, such as pure P2P networks, ad hoc and sensor networks, grids and so on. Many searching algorithms for decentralized networks have been designed[9], all of which perform three basic actions: searching of active nodes, querying for resources or services, and content transferring.

3) Partially Centralized Networks (also known as hybrid networks, layered networks or multi-tier networks). In these networks, there are some nodes with higher roles which form the backbone of the network and are needed to run the system. Nodes with the lower role are called *leaf nodes* and will be placed in the lower logical layer, while nodes with the higher roles could be *supernodes* and will be placed in the higher logical layers. Every supernode or leaf node can have connections with either the other leaf nodes or supernodes. There is a hierarchy where higher layer nodes organize, control or gather data from lower layer nodes. The higher layer nodes are used for forwarding the messages from the lower layer nodes. Layered networks have been used for different types of networks such as satellite networks[10], wireless networks[11] and even models for business processes[12].

Let us suppose we need to divide the network into groups or areas according to the physical implementation of the WAHSN or for scalability purposes. It does not matter which kind of routing protocol is being used inside each group. All architectures shown above fail to solve that problem efficiently, because in the case of centralized architectures, the server will have many wireless connections at the same time, so it will need many resources. There is also a central point of failure and a bottleneck. On the other hand, in the case of fully distributed architectures, it is very difficult to control the system and it needs a long time to process tasks (because of the time needed to reach far nodes), decreasing the performance of the system.

We propose dividing the whole WAHN or wireless sensors and actor networks (WSAN) into several groups, and that when a node receives data for its group, it will propagate the data to the rest of the nodes in its group.

A group is defined as a small number of interdependent nodes with complementary operations that interact in order to share resources or computation time, or to acquire content or data and produce joint results. In a wireless group-based architecture, a group consists of a set of nodes that are close to each other (in terms of geographical location, coverage area or round trip time) and neighboring groups could be connected if a node of a group is close to a node of another group. The main goal in a wireless group-based topology is the network protocol and the group management, that is, the design of an efficient algorithm and a capable protocol is needed to find the nearest (or the best) group to join in when a new node appears in the network. The performance of the network largely depends on the efficiency of the nearby group locating process and on the interaction between the neighbor groups.

We have to distinguish between a groupware architecture and a group-based architecture. In a groupware architecture all nodes collaborate towards the correct operation and the success of the network purpose, while in a group-based architecture the whole network is broken down into groups and each group can perform different operations or can have different routing protocols.

Some important issues must be taken into account in a wireless group-based architecture regardless of the protocol inside the group as follows.

1) How to build neighboring groups.

2) A protocol to exchange messages between neighboring groups.

We can distinguish two types of group-based topologies: planar group-based topologies and layered group-based topologies. In planar group-based topologies all nodes perform the same roles and there is only one layer. However, in some work there is a directory server or a rendezvous point (RP) for content distribution coordination. Nodes from layered group-based topologies could have several roles (2 roles at least). Depending on which type of role they are playing, they will belong to a specific layer. All nodes in the same layer will have the same role. There will be connections between nodes from the same layer and from different layers, but these layers must be adjacent. We have included hierarchical architectures in this group, because the hierarchies could be considered as layers. There are several differences between both the group-based topologies. While layered group-based topologies grow in a structured form, organized by upper layers, planar group-based topologies grow in an unstructured form, without any organization. On the one hand, in layered group-based topologies any node can know exactly where each group is and how to reach it; on the other hand, planar group-based topologies, because the groups join the network as they appear, and every time there is a connection between the nodes from different groups, the message should travel through many unknown groups in the path. Delays between groups in layered group-based topologies could be lower because connections between groups can be established taking this parameter into account. In planar group-based topologies, connections between groups are established by the group's position, their geographical situation or their appearance in the network. Layered networks involve some complexity because nodes could have several types of roles and fault tolerance must be designed for each layer. Planar networks are simpler because all nodes have the same role. In order to be more scalable, layered group-based topologies must add more layers to its logical topology, while planar group-based topologies could grow without any limitation, just the number of hops of the message.

Group-based networks provide some benefits for the whole network, such as the following.

- Spread the work efficiently to the network in groups, giving more flexibly, and lower delays.
- Content availability will increase because it could be replicated in other groups.
- Anyone could search and download data from every group using only one service.
- Fault tolerance. Other groups could carry out tasks from a failed one.
- Scalability. A new node can join any group and a new group could be added easily.
- Network measurements could be taken from any

group.

There are some works in the literature where nodes are divided into groups and connections are established between nodes from different groups, but all of them have been developed to solve specific issues[13−16], but none of them for MANET networks.

The Rhubarb system[13] organizes nodes in a virtual network, allowing connections across firewalls/NAT (Network Address Translation), and efficient broadcasting. Nodes can be active, if they establish connections, or passive, if they do not do it. The Rhubarb system has only one coordinator per group and coordinators could be grouped hierarchically. It uses a proxy coordinator, an active node outside the network, and all nodes inside the network make a permanent TCP connection with the proxy coordinator, which, if broken, can be renewed by the firewall or NAT. When a node from outside the network wishes to communicate with an inner node, it sends a connection request to the proxy coordinator, which forwards the request to the inner node.

A Peer-to-Peer Based Multimedia Distribution Service was presented in [14]. Xiang *et al.* proposed a topology-aware overlay in which nearby hosts or peers self-organize into application groups. End hosts within the same group have similar network conditions and can easily collaborate with each other to achieve Quality of Service (QoS) awareness. When a node wants to communicate with a node from another group, the information is routed through several groups until it reaches its destination.

There are some hierarchical architectures where nodes are structured hierarchically and parts of the tree form groups, such as the ones in references [15, 16]. In some cases, some nodes have connections with nodes from other groups although they are in different layers of the tree, but in all cases, the information has to be routed through the hierarchy.

There are many cluster-based hierarchical architectures[17]. In a cluster-based architecture the mobile nodes are divided into virtual groups. Each cluster has adjacencies with the other clusters. All the clusters have the same rules. A cluster can be made up of a Cluster Head node, Cluster Gateways and Cluster Members[18,19]. The Cluster Head node is the parent node of the cluster, which manages and checks the status of the links in the cluster, and routes the information to the right clusters. The rest of the nodes in a cluster are all leaf nodes. In this kind of network, the Cluster Head nodes have a total control over the cluster and the size of the cluster is usually about 1 or 2 hops from the Cluster Head node. The cluster gateways have links to other clusters and route the information to those clusters. On the other hand, a cluster member is a node without any inter-cluster links. Finally, we want to emphasize that the cluster-based networks are a subset of the group-based networks, because every cluster could be considered as a group. But a group-based network is capable of having any type of topology inside the group, not only clusters. However, both types of networks have been created for solving the scalability problems of the WAHSN.

We can also find in the literature a routing protocol based on zones. It is the Zone Routing Protocol (ZRP)[20,21]. Each node proactively maintains routing information for a local neighborhood (routing zone), while reactively acquiring routes to destinations beyond the routing zone. ZRP and our proposal have several common features, e.g., they could be applied over any type of routing protocol, they scale well and the information is sent to border nodes in order to reach destinations outside their zones. The main difference between them is that in ZRP each node maintains a zone and the nodes in that zone have different nodes in their zone while in our proposal all the nodes that form a group have the same nodes in their group.

On the other hand, we will not consider other work of group systems such as the following. The community-based mobility model for ad hoc network research presented in [22], because although the network is organized in groups, and nodes can move from one host to another, there is not any connection between border nodes from different groups. The landmark hierarchy presented in [23], because although there is a node with a higher role which has connections with the nodes from the other groups, its leaf nodes do not. Another example similar to the last one is the BGP routing protocol architecture[24]. Finally, we will not consider moving groups such as Landmark Routing Protocol (LANMAR[25]), where the set of nodes move as a group, so the group can enlarge or diminish with the motion of the members.

3 Application Environment

Group-based networks can be used when there is a need to setup a network where groups could appear and join the network at anytime or when the network has to be split into smaller zones to support a large number of nodes, that is, in any system where the devices are grouped and there must be connections between groups.

The following list gives several group-based WAHSN application areas.

1) Let us suppose a job where all human resources need to be split into groups to achieve a purpose (such as fire fighter squads for putting out the fire). Now, let

us suppose that all the people involved in that activity need a device that has to be connected with other devices in the same group to receive information from the members within the group, and closer groups have to be connected to coordinate their efforts. Currently coordination between groups is done through a wireless connection to the command center or using satellite communications. But, some times neither of those solutions can be used because a free obstacle line of sight is needed, because there are too many wall looses or because more gain or power is needed to reach the destination.

2) For battle field communication, it is especially useful for inter-squad communication to collaborate when an objective is targeted by position detectors.

3) Groups could also be established because of geographical locations or unevenness. It happens in rural and agricultural environments. A group-based topology in this kind of environment could be useful to detect plagues or fire and to propagate an alarm to neighbor lands. It will provide easier management and control for detecting fires and plagues as well as for allowing scalability.

4) Health monitoring[26]. A patient might need to be monitored in several locations while he is doing his activity. Every room or place could have a group of sensors (and even each group with different type of topology inside) and neighbor groups must be communicated to keep track of the patients.

5) It could be used in any kind of system in which an event or alarm is based on what is happening in a specific zone, but conditioned to the events that are happening in neighbor zones. One example is a group-based system that measures the environmental impact on a place. It could be better measured if the measurements are taken from different groups of sensors, but those groups of sensors have to be connected in order to estimate the whole environmental impact.

6) Group-based virtual games. There are many games where the players are grouped virtually in order to perform a specific task. Interactions between groups in virtual reality should be given by interactions between players from different groups to exchange their knowledge.

In the following section we will show that group-based topologies give better performance to the whole wireless ad hoc and sensor network.

4 Group-Based WAHSN Topologies Performance

This section compares the performance of 3 common MANET protocols and shows which one is the best when they are using group-based topologies.

4.1 Test Bench

First, we present the test-bench used for all the evaluated protocols. The number of nodes and the coverage area of the network have been varied. We have simulated 4 scenarios for each protocol: the first one with fixed nodes; the second one with mobile nodes and failures; the third one with grouped nodes; and, the fourth one with grouped mobile nodes and failures. We have simulated each scenario for 100 and 250 nodes to observe the system scalability. It has been obtained using the version Modeler of OPNET simulator[27].

Instead of a standard structure we have chosen a random topology. The nodes can move randomly during the simulation. The physical topology does not follow any known pattern. The obtained data do not depend on the initial topology of the nodes nor on their movement pattern, because all of it has been fortuitous.

In order to take measurements from the mobile nodes simulation, we have forced failures in the networks with the consequent recovering processes. It allows us to observe the network behavior, against physical topology changes and node failures. Failures and recoveries usually happen in these kinds of networks, so, we are going to study how a network-level protocol works when those events occur.

We have created 6 groups for the 100 nodes topology, covering approximately, a circular area with a 150 meter radius each group. There are approximately 16 or 17 nodes in each group. The number of nodes in each group varies because of the node's random mobility. A node can change a group anytime. For the 250 nodes topology, we have created 12 groups, with 15 or 16 nodes per group approximately, covering a circular area with a 150 meter radius each group.

The ad-hoc nodes of the topologies have a 40MHz processor, a 512KB memory card, a radio channel of 1Mbps and their working frequency is 2.4GHz. Their maximum coverage radius is 50 meters. This is a conservative value because most of the nodes in ad-hoc network have larger coverage radius, but we preferred to have lower transmitting power for the ad-hoc devices to enlarge their lifetime.

The traffic load used in the simulations is MANET traffic generated by OPNET. We inject this traffic 100 seconds after the beginning. The traffic follows a Poisson distribution (for the arrivals) with a mean time between arrivals of 30 seconds. The packet size follows an exponential distribution with a mean value of 1024 bits. The injected traffic has a random destination address, obtaining a simulation independent of the traffic direction. We have simulated both scenarios for DSR, AODV and OLSR protocols. The results obtained are shown in the following subsections.

4.2 Average Delay at Application Layer

Figs.1 and 2 show the average delay of the DSR protocol in fixed and mobile topologies at the application layer. In Fig.1 we observe that group-based topologies have an average delay close to 0.005 seconds regardless of the number of nodes in the network. In the regular network the delay has a value of 0.02 seconds for 100-node topology and of 0.03 seconds for the 250-node topology when the network converges. In the case of the 100-node topology there is an improvement of 75%, and it is better in the 250-node topology (an 83% improvement). The topologies with mobility and errors (Fig.2) show that the average delays at the application layer are higher in the group-based topologies until the network converges. We observe that group-based topologies present worse behavior up to 1300 seconds. Then, the delay decreases. There is an improvement of around 5%.

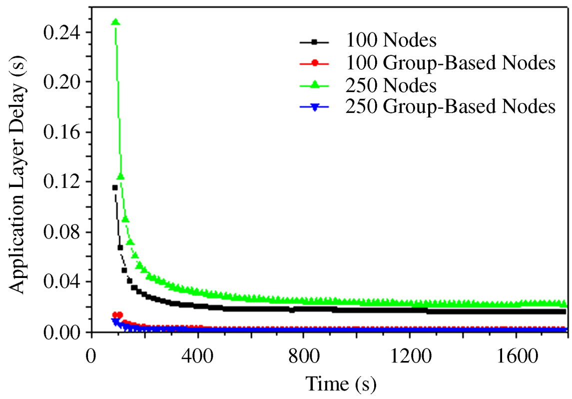


Fig.1. DSR average delay at the application layer in fixed topologies.

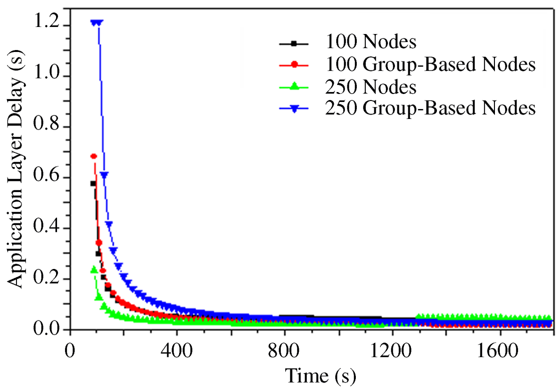


Fig.2. DSR average delay at the application layer in mobile topologies.

The average delay at the application layer in the AODV protocol can be seen in Figs.3 and 4. When we are talking about fixed topologies (Fig.3), both of 100-node and 250-node, give an average delay higher than 0.5 seconds when the network converges, but there are some peaks higher than 2.5 seconds. On the other hand, group-based topologies have a similar delay which is around 0.15 seconds. Group-based topologies improve the delay at the application layer by 70%. When the topology with mobile nodes is used, the simulation shown in Fig.4 is obtained. In the case of 250 nodes, there is a delay of 1 second when the network has converged. The case of 100 nodes gives an average delay around 0.75 seconds. When there are group-based topologies, the delay decreases to 0.25 seconds in both cases. There is an improvement of 75% for the 250-node topology and 67% for the 100-node topology.

In Fig.5, the delay at the application layer for the OLSR protocol using fixed topologies is shown. In the case of 250 nodes we have obtained a delay of around

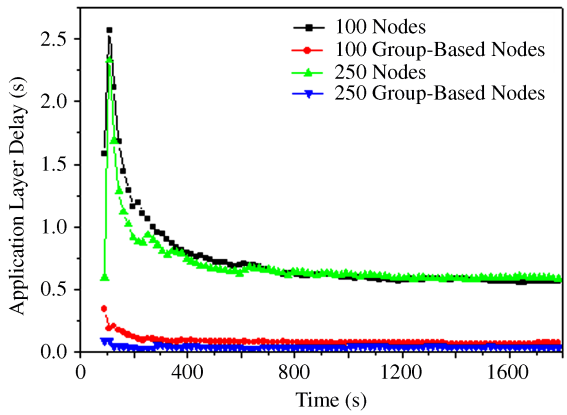


Fig.3. AODV average delay at the application layer in fixed topologies.

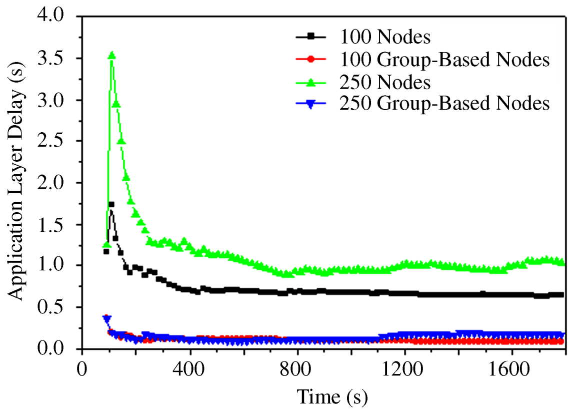


Fig.4. AODV average delay at the application layer in mobile topologies.

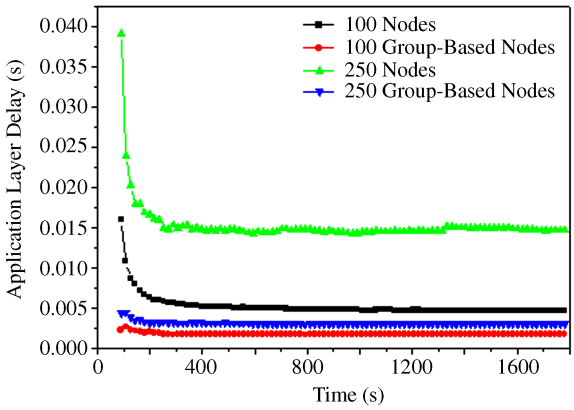


Fig.5. OLSR average delay at the application layer in fixed topologies.

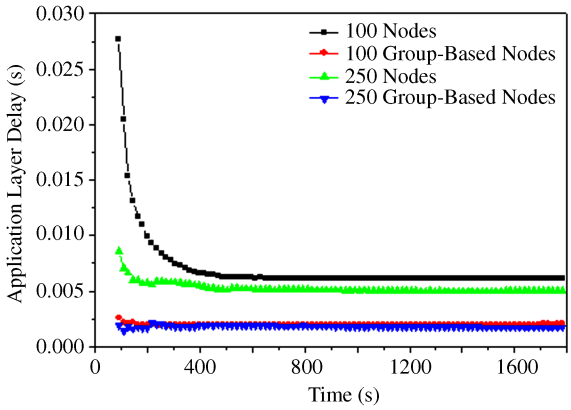


Fig.6. OLSR average delay at the application layer in mobile topologies.

0.015 seconds, and the delay has changed to 0.0035 seconds in the case of 250-node group-based topology (there is a 76% improvement). In the case of 100 nodes, the delay has decreased from 0.005 seconds in the regular topology to 0.002 seconds in the group-based topology, so there is a 60% improvement. When there is mobility, errors and failures in the network for the OLSR protocol (see Fig.6), we observe that the 100-node regular topology has a delay at the application layer of 0.007 seconds when the network has converged, but there is a delay of 0.0025 seconds for the 100-node group-based topology (a 64% improvement). In the case of 250 nodes the improvement is around 60%. We have obtained a delay of 0.005 seconds in the regular topology versus 0.002 seconds in the group-based topology.

4.3 Routing Traffic Received

We have compared the routing traffic received in the DSR protocol (Figs.7 and 8). Fig.7 shows that the traffic is quite stable because it is a fixed network without errors or failures. The traffic received in the 250-node topology is around 500Kbits/s, but when we group the nodes, this traffic decreases to 200Kbits/s (a 60% improvement). The value obtained in a 100-node topology (250Kbits/s) is also improved when we group the nodes (100Kbits/s), therefore there is a 60% improvement. In Fig.8 we observe a similar behavior. In this case we conclude that when there are errors and failures in the 250-node topology the traffic fluctuates and is less stable (we can observe it in the intervals from 600 to 800 seconds and around 1200 seconds). We also observe that the instability is much lower in group-based topologies. 100-node topology has a mean value around 175Kbits/s, while 100-node group-based topology has a mean value around 95Kbits/s, so there is an improvement of 46%. On the other hand, 250-node topology has a mean value around 400Kbits/s, while 250-node group-based topology has a mean value around 180Kbits/s, so there is an improvement of 55%.

Then, the routing traffic received for the AODV in each simulated topology can be seen in Figs.9 and 10.

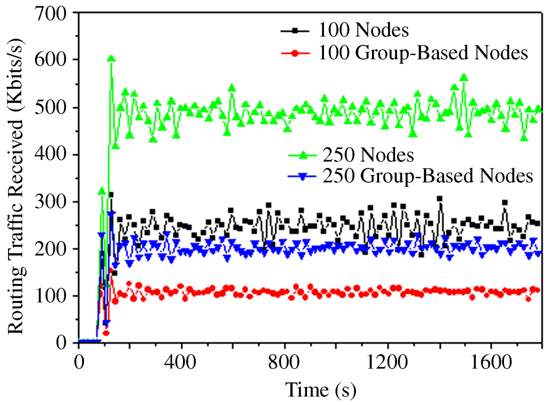


Fig.7. DSR routing traffic received in fixed topologies.

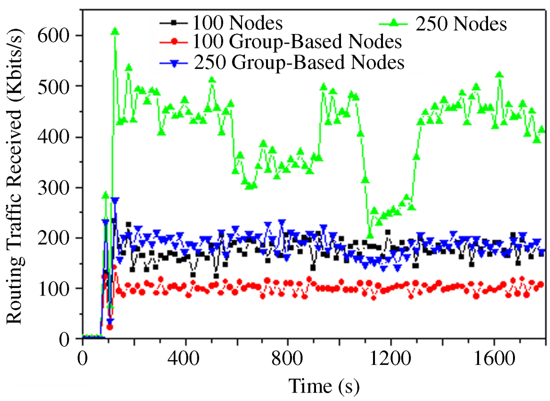


Fig.8. DSR routing traffic received in mobile topologies.

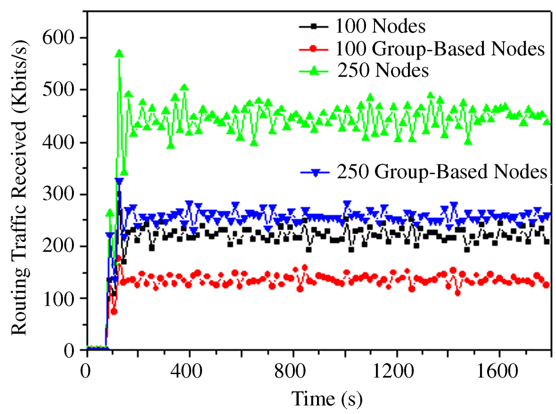


Fig.9. AODV routing traffic received in fixed topologies.

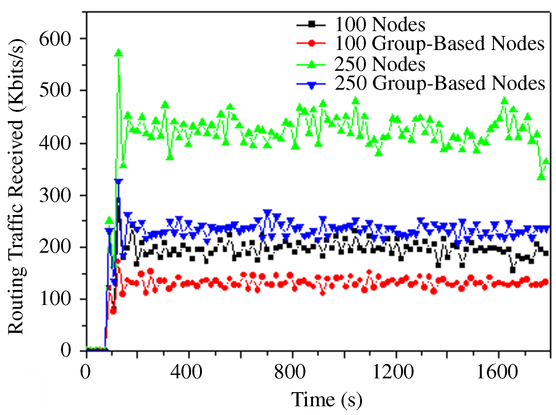


Fig.10. AODV routing traffic received in mobile topologies.

We observe that the routing traffic received is independent of the mobility of the nodes. In Fig.9 we can see that the routing traffic goes from 440Kbits/s for 250-node case to 250Kbits/s when there are groups of nodes (a 43% improvement). In the 100-node topology, it goes from 230Kbits/s to 140Kbits/s in the group-based topology case (a 39% improvement). When there are mobility, errors and failures (see Fig.10), in the 250-node topology the values go from 440Kbits/s to 250Kbits/s in the group-based topology (a 43% improvement). We obtained 200Kbits/s in the regular 100-node topology and 135Kbits/s for the group-based one (a 32% improvement).

Finally, we have studied the behavior of the OLSR protocol analyzing the mean routing traffic received (Figs.11 and 12). In Fig.11, we see that the routing traffic received in the 100-node fixed topology is around 180Kbits/s, while in group-based topology it has decreased to 70Kbits/s, so there is a 61% improvement. In the 250-node topology case, we appreciate that this traffic was approximately 300Kbits/s, but there are values lower than 150Kbits/s in the group-based topology,

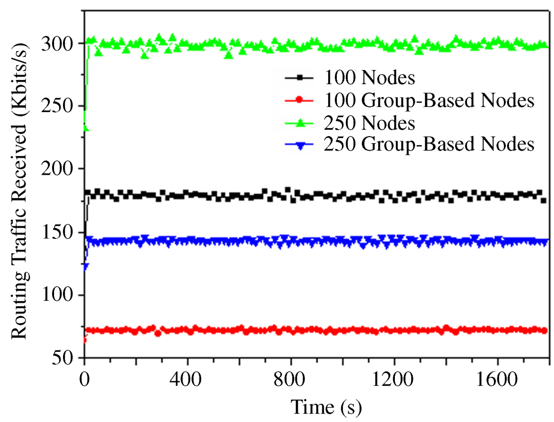


Fig.11. OLSR routing traffic received in fixed topologies.

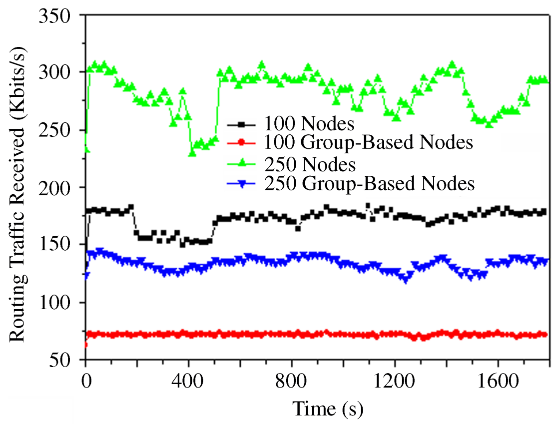


Fig. 12. OLSR routing traffic received in mobile topologies.

so there is a 50% improvement. Fig.12 shows the results of a network with mobility and errors and failures. We have observed some fluctuations due to the failures and errors in the network, in both 100-node and 250-node topologies. Those fluctuations are minimized when we use group-based topologies. Improvements of 61% and 50% are obtained in 100-node and 250-node topologies, respectively.

4.4 Throughput

When we study the network throughput (Figs.13 and 14), we observe that group-based topologies give a much lower value than the one obtained in regular topologies. For the 100-node topology (Fig.13), the throughput varies from 225Kbits/s to 100Kbits/s in the group-based topology (a 56% improvement). In the 250-node topology we obtain 460Kbits/s of throughput for the regular topology and 190Kbits/s of throughput for the group-based one (a 59% improvement). Moreover, when we compare Figs.13 and 14, we can conclude that the throughput in group-based topologies has a very low variation regarding a fixed or mobile scenario. The obtained improvement is quite important. We can see in Fig.14 that, after 1200 seconds, the obtained throughput in 250-node topology is similar to the obtained throughput in the 100-node topology.

Fig.15 shows the throughput for fixed topologies. The 100-node scenario gives a 200Kbits/s mean value,

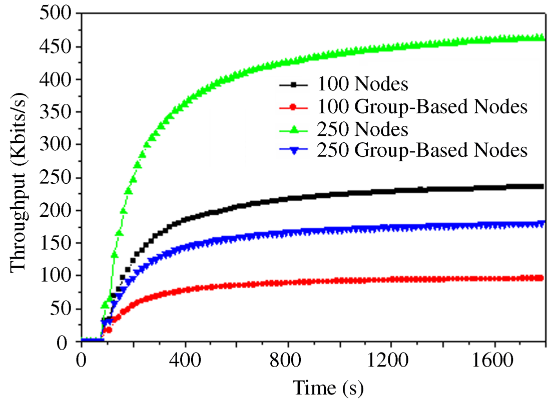


Fig.13. DSR mean throughput in fixed topologies.

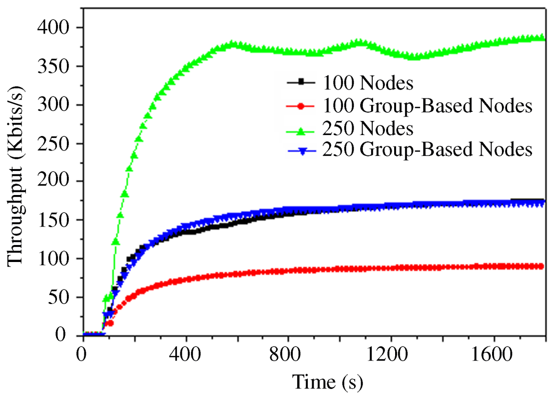


Fig.14. DSR mean throughput in mobile topologies.

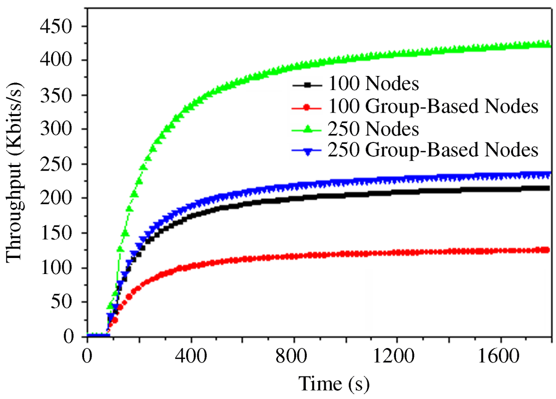


Fig.15. AODV mean throughput in fixed topologies.

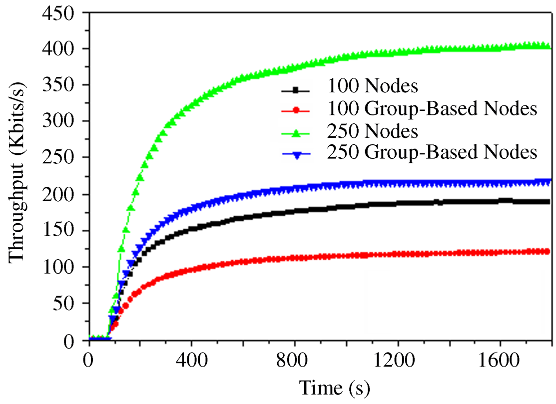


Fig.16. AODV mean throughput in mobile topologies.

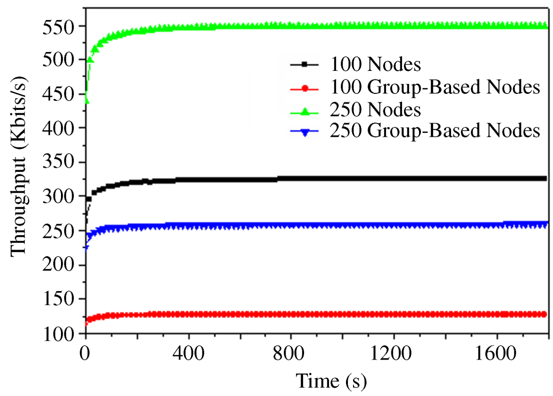


Fig.17. OLSR mean throughput in fixed topologies.

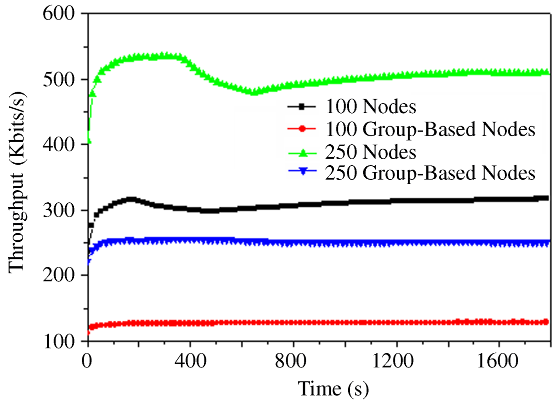


Fig.18. OLSR mean throughput in mobile topologies.

but a value of 120Kbits/s is obtained for the group-based scenario (a 40% improvement). In the 250-node case, we obtain mean values of 425Kbits/s for the fixed scenario and of 225Kbits/s for the group-based scenario (a 47% improvement). Fig.16 shows the results for mobile topologies with errors and failures. The improvement obtained by grouping nodes decreases in the 100-node case (37%), but it does not vary in the 250-node cases.

Finally, the mean throughput measured in fixed topologies can be observed in Fig.17. In scenarios with 250 nodes we obtained a mean throughput of 550Kbits/s and 250Kbits/s (group-based, with a 54% improvement). In 100-node regular topology the throughput is 325Kbits/s and 125Kbits/s (group-based, with a 61% improvement). When we consider mobility, errors and failures (Fig.18) the throughput is not so stable as in above case but, we can observe that the improvements are quite similar. In the case of 250 nodes we obtain a 52% improvement in the group-based scenario; in the case of 100 nodes the improvement reaches the 60%.

4.5 Group-Based Topologies Comparison

In order to make the comparison of DSR, AODV and OLSR using group-based topologies, we have used the same test bench used previously. This comparison will show us which mobile and ad-hoc routing protocol performs better using group-based topologies.

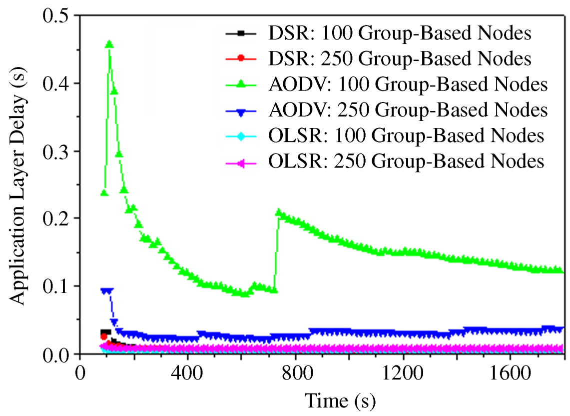


Fig.19. Comparison of the average delay at application layer in fixed topologies.

Fig.19 shows the average delay at application layer in fixed group-based topologies. The most instable protocol with higher delay in 100-node and 250-node topologies is AODV protocol. It has peaks with more than 0.45 seconds and it is stabilized around 1700 seconds with a mean value of 0.15 seconds. DSR and OLSR are the ones with lowest delay. Fig.20 shows the average delay at application layer in mobile group-based topologies. DSR protocol is the one that has the worst delay until the network converges. Then, when the network is stabilized, the worst is AODV protocol which has delays between 0.1 and 0.15 seconds. OLSR protocol gives the lowest delays.

The routing traffic received in fixed and mobile group-based topologies is shown in Figs.21 and 22, respectively. In fixed group-based topologies (see Fig.21) AODV protocol is the one that gives higher routing traffic received (around 250Kbits/s in 250-node topology and 135Kbits/s in 100-node topology). OLSR protocol is the most stable and the one with lower routing traffic received (145Kbits/s in 250-node topology and 70Kbits/s in 100-node topology). When the mobile group-based topologies are analyzed (Fig.22), AODV protocol is the one that has the worst behaviour and OLSR is the most stable and the one that has lower routing traffic sent. DSR protocol is the most instable.

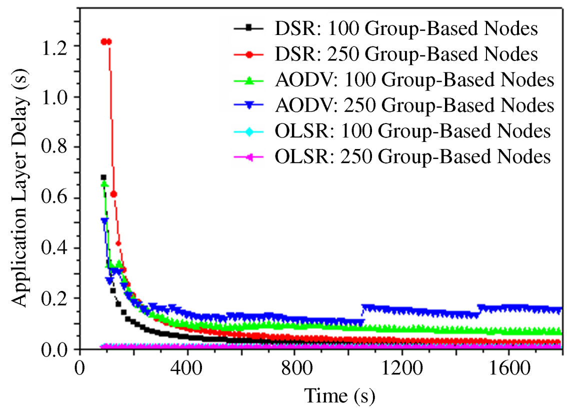


Fig.20. Comparison of the average delay at application layer in mobile topologies.

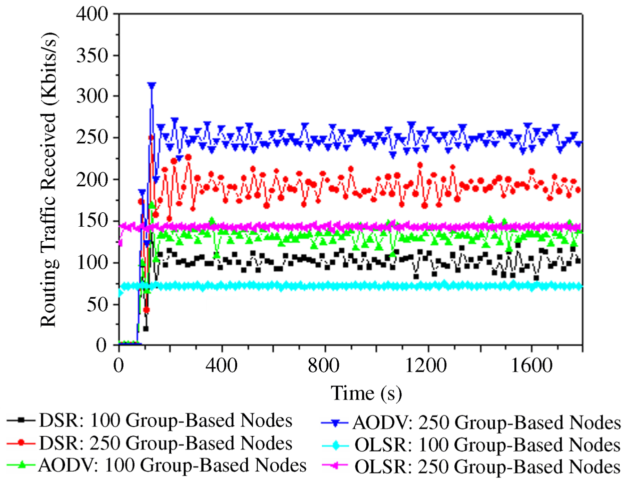


Fig.21. Comparison of routing traffic received in fixed topologies.

The average throughput consumed in the fixed group-based topologies is compared in Fig.23. The protocol that consumes the lowest throughput is the DSR protocol (90Kbits/s in the 100-node topology and 170Kbits/s in the 250-node topology). The protocol with the most stable throughput consumed is the OLSR

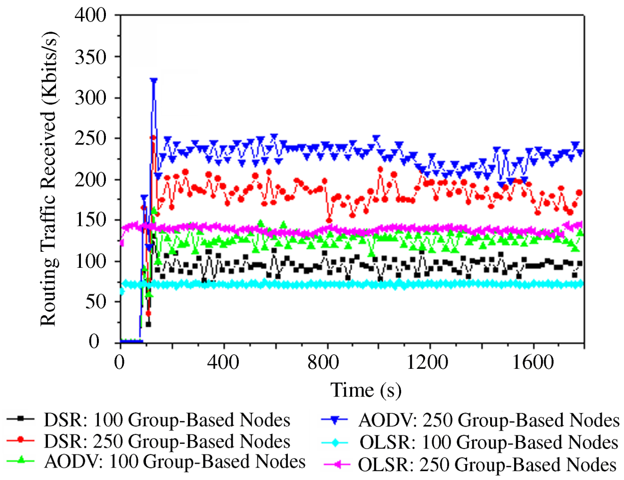


Fig.22. Comparison of routing traffic received in mobile topologies.

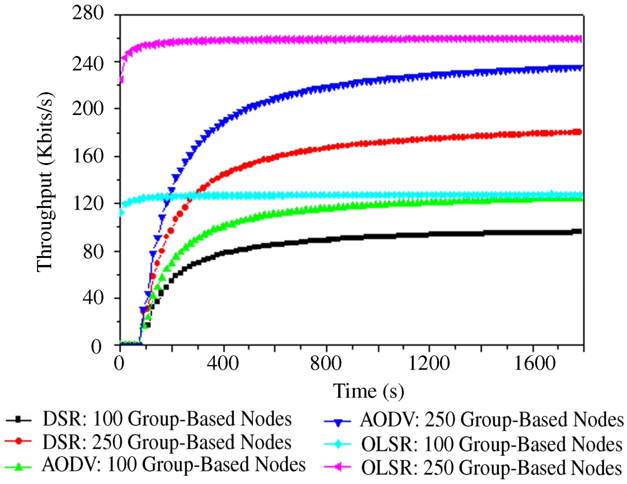


Fig.23. Comparison of average throughputs consumed in fixed topologies.

protocol. When the network converges, both AODV and OLSR protocols have the same average throughput in the 100-node topology, but the OLSR protocol has the lowest convergence time.

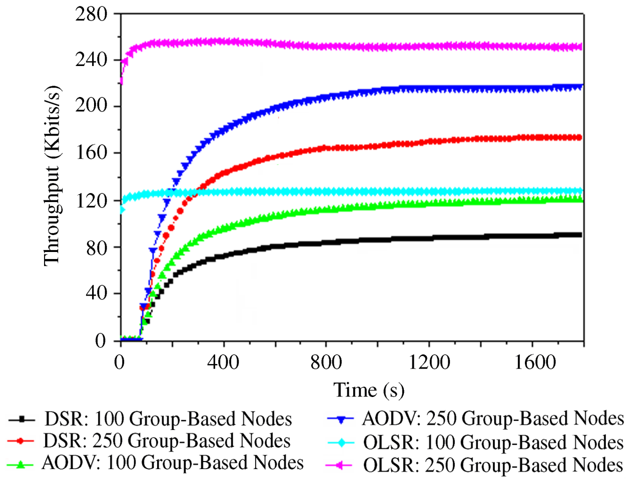


Fig.24. Comparison of average throughputs consumed in mobile topologies.

In case of having a group-based topology with mobility, errors and failures (see Fig.24), the results are very similar to the previous ones. The protocol that consumes lower throughput is DSR. AODV protocol consumes lower throughput while the network is converging, but this throughput becomes very similar to the one given by OLSR protocol when the network converges. OLSR protocol is still the most stable.

4.6 Analyzed Protocols Summary

In this subsection we show the benefits of using a group-based topology in ad-hoc networks, and we show several examples in which they can be used. We have simulated DSR, AODV and OLSR protocols with and without groups and the results show that group-based topologies give better performance.

In Table 1 we can see a summary where there is percentage improvement when group-based topologies are used.

Table 1. Percentage of Improvement When Group-Based Topologies Are Used

	Fixed Topology (100 Nodes)	Fixed Topology (250 Nodes)	Mobile Topology (100 Nodes)	Mobile Topology (250 Nodes)
DSR Average Delay at the Application Layer	75%	83%	5%	5%
DSR Routing Traffic Received	60%	60%	46%	55%
DSR Mean Throughput	56%	59%	48%	55%
AODV Average Delay at the Application Layer	70%	70%	67%	75%
AODV Routing Traffic Received	39%	43%	32%	43%
AODV Mean Throughput	40%	47%	37%	47%
OLSR Average Delay at the Application Layer	60%	76%	64%	60%
OLSR Routing Traffic Received	61%	50%	61%	50%
OLSR Mean Throughput	54%	61%	52%	60%

Table 2. Comparison of Mobile and Ad-Hoc Routing Protocols in Group-Based Topologies

	Best in Fixed	Best in Mobile	Worst in Fixed	Worst in Mobile
Delay at MAC Layer	OLSR	OLSR	DSR	AODV
Throughput Consumed	DSR	DSR	AODV & OLSR	AODV & OLSR
MANET Traffic	AODV	DSR	OLSR	OLSR
Routing Traffic Sent	OLSR	OLSR	AODV	AODV
Routing Traffic Received	OLSR	OLSR	AODV	AODV
Delay at Application Layer	DSR & OLSR	OLSR	AODV	AODV
Average Number of Hops in a Path	AODV	DSR	AODV	DSR
Route Request Sent	DSR	AODV	DSR	AODV

In this study we have made other measures. Table 2 shows the best and worst protocols for every one of the parameters analyzed.

The best improvement percentage, when group-based topologies were used, came from the DSR protocol when the average delay at the application layer was simulated. On the other hand, in the same case for mobile topologies, DSR protocol gave the worst percentage of improvement.

We observed it has more percentage of improvement in fixed topologies when there are more nodes in the topology, but when there is a mobile topology, the improvement is higher in the topology with lower number of nodes. We have also observed that when a routing protocol is the best one in a fixed group-based topology, it continues being the best one in the mobile group-based topology. On the other hand, we observed that a routing protocol, which is the best (or worst) in a group-based fixed topology, could not be the best (or worst) in the mobile topology. The routing protocol that appeared as the best one was OLSR and the one that appeared as the worst was AODV.

5 Architecture Description

5.1 Architecture Operation

We propose an architecture of nodes and a protocol based on the creation of groups of nodes where nodes have the same functionality in the network. Every group has a central node that limits the zone where the node from the same group will be placed, but its functionality is the same as the rest of the nodes. Every node has a *nodeID* that is unique in its group. The first node in the network acquires a group identifier (*groupID*) that is given manually, using GPS (Global Positioning System), or using a wireless location system or through other means[28]. New joining nodes will know their group identifier from their new neighbors. Border nodes are, physically, the edge nodes of the group. When there is an event in a node, this event is sent to all the nodes in its group in order to take appropriate actions. All nodes in a group know all the information about their group. Border nodes have connections with other border nodes from neighbor groups and are used for sending information to other groups or receiving information from other groups and distributing it inside. Because a fast routing protocol is needed, we have chosen SPF (Shortest Path First) routing algorithm[29] to route information, but it can be changed by the other routing protocols depending on the network's characteristics. When the information is for a node of the same group it is routed using the *nodeID*. Every node runs SPF algorithm locally and selects the best path to a destination based on a metric. But, when the information has to be sent to other groups, the information is routed directly to the closest border node to the destination group using the *groupID*. When a node from a destination group receives the information, it routes it to all nodes in its group using Reverse Path Forwarding Algorithm[30]. Links between border nodes from different groups are established primarily as a function of their positions, but, in the case of multiple possibilities, neighbors are selected as a function of their capacity λ which will be explained in the following section. In order to establish the boundaries of the group, we can consider two choices: (i) limiting the diameter of the group to a maximum number of hops (e.g., 30 hops, as the maximum number of hops for a tracer of a route), and (ii) establishing the boundaries of the area that is to be covered. Fig.25 shows the proposed architecture topology.

5.2 Analytical Model and Neighbor Selection

Every node has 3 parameters (*nodeID*, *groupID* and λ) that characterize the node. Let λ parameter be the node capacity that depends on the node's upstream and downstream bandwidth (in Kbps), its number of available links (*Available_Con*) and its maximum number of links (*Max_Con*), its percentage of available load and its energy consumption. It is used for determining the best node to connect with. The higher the λ parameter,

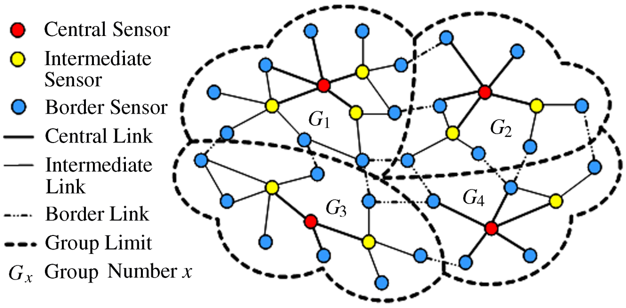


Fig.25. Proposed architecture topology.

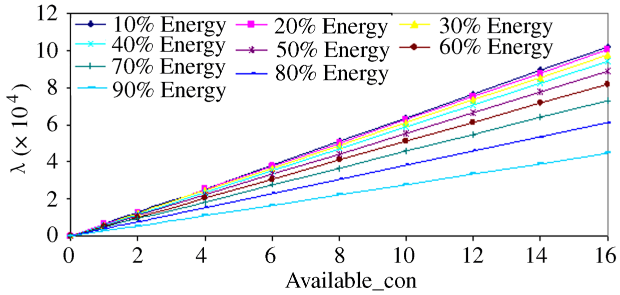


Fig.26. λ parameter values with number of links variation.

the better node to connect with. λ equation is shown in (1)

$$\lambda = \frac{(BW_{\rm up} + BW_{\rm down}) \cdot Available\_Con \cdot L + K_2}{Max\_Con} \cdot \sqrt{1 - \frac{E^2}{K_1}}, \quad (1)$$

L is the available load and E is the energy consumption. Their values vary from 0 to 100. $E = 0$ indicates it is fully charged, so λ parameter is 0 and $E = 100$ indicates it is fully discharged.

K_1 defines the minimum value of energy remaining in a node to be suitable for being selected as a neighbor. K_2 gives different λ values from 0 in the case of $L = 0$ or $Available\_Con = 0$. We have considered $K_2 = 100$, to get λ into desired values. Fig.26 shows λ parameter values at the time when the maximum number of links for a node is 16, for a bandwidth value of 2Mbps, as a function of its available number of links for different available energy values of the node. Node's load is fixed to 50%. Fig.27 shows λ parameter values when the maximum number of links of the node is 16 as a function of the node energy available for different bandwidth values. Node's load is fixed to 80% and all nodes have 6 available number of links ($Available\_con$ = 6). It shows that as the Energy is being consumed, λ parameter is lower, but when it gets the 80% of consumption, the λ parameter decreases drastically, so the node is more likely to be chosen as a neighbour, in case of more energy available. Fig.27 also shows that a node with higher bandwidth is preferred.

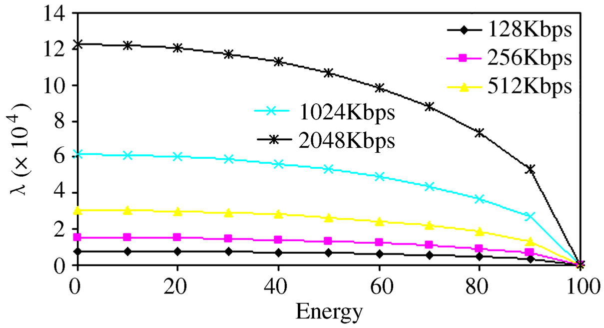


Fig.27. λ values as a function of the Energy of the node.

We have defined the cost of the i-th node as the inverse of the i-th node parameter multiplied by T (the delay of its reply in ms). The cost is shown in (2)

$$C = \frac{T \cdot K_3}{\lambda}. \quad (2)$$

$K_3 = 10^3$ gives $C \geqslant 1$. The metric for each route is based on the hops to a destination (r) and on the cost of the nodes (C_i) in the route as shown in (3)

$$metric = \sum_{i=1}^{r} C_i. \quad (3)$$

The metric gives the best path to reach a node.

Let $G = (V, \lambda, E)$ be a network of nodes, where V is the set of nodes, λ is the set of their capacities ($\lambda(i)$ is the capacity of the i-th node and $\lambda(i) \neq 0$ $\forall i$-th node) and E is the set of links between nodes. Let k be a finite number of disjoint subsets of V, so $V = \cup V_k$, and there is no node in two or more subsets ($\cap V_k = 0$), and let $n = |V|$ (the number of nodes in V), the equation given for n is shown in (4)

$$n = \sum_{i=1}^{k} |V_k|. \quad (4)$$

Every V_k has a central node, several intermediate nodes and several border nodes as shown in (5)

$$n = 1 + n_{\rm intermediate} + n_{\rm border}. \quad (5)$$

Now we can describe the whole network as the sum of all these nodes from all groups as shown in (6)

$$n = \sum_{i=1}^{k} |(n_{\rm central} + n_{\rm intermediate} + n_{\rm border})_k| = k + \sum_{i=1}^{k} (|n_{\rm intermediate}|)_k + \sum_{i=1}^{k} (|n_{\rm border}|)_k. \quad (6)$$

On the other hand, the number of links in the whole network $m = |E|$ depends on the number of groups (k), on the number of links in each group (k_m) and on the number of links between border nodes. (7) gives m value for a physical topology.

$$m = \sum_{i=1}^{k}\left(k_l + \frac{1}{2}k_b\right) \tag{7}$$

where k_l is the number of links inside the group k and k_b is the number of external links of the group k.

6 Protocol Operation and Messages

This section describes the designed messages and how the designed protocol operates.

6.1 Group Creation and Maintenance

Let a new node join the network (it could be the first). It sends a hello message (called *helloGroup*) in order to join a group. If there is no response from any node for 3 seconds, the node considers itself as a central node of a group in the network, and it will take the value *groupID* = 1 and *nodeID* = 1. When the node receives *helloGroup ACK* messages from several candidate neighbors, first it puts a timestamp on their reply and chooses the best nodes to have a link with (this election is taken based on the λ parameter which comes in the *helloGroup ACK* message). The timestamp will be used to calculate C parameter. Responses received after 3 seconds will be discarded. In case of receiving replies from nodes of different groups, it will choose the group whose replies have the highest average λ parameter, so it will take into account replies only from that group. Then, the node will send an *okGroup* message to the selected neighbors, and the neighbors will reply with the *okGroup ACK* message with the assigned *nodeID* and indicates the link has been established. Nodes will send *keepalive* messages periodically to their neighbors. If a node does not receive a *keepalive* message from a neighbor before the dead time, it will remove this entry from its database and will start the group update process. As the *groupID* is in the *helloGroup ACK* message, the new node will know which group has joined. Finally, the neighbor node will send a *newNode* message to the central node, to run the algorithm for changing the central node if needed.

Links between border nodes from different groups are established as a function of their replying delay and the λ parameter of the replying nodes, but it could be changed by an algorithm using node's position or choosing the neighbor with the shortest distance (in number of hops) to the central node. If we base our proposal on the λ parameter, we will distribute the load of the network between groups, but if we base our proposal on a node's position or choose the neighbor with the shortest distance to the central node we will balance the number of nodes in the groups.

When a new node joins the group, the central node of the group could be changed. The procedure designed for changing the central node is as follows. We define the group diameter (d_{group}) as the smallest number of hops, between the two most remote nodes in the group (in our case, $d_{\text{group}} \leqslant 30$).

When there is a change of the central node of a group, all the nodes in the group must be alerted. In order to update all nodes in the group, the new central node will send a *changeCentral* message to indicate the new central node and the distance from it to the node processing this control packet. This update is distributed using the Recursive Proportional-Feedback (RPF) algorithm. Once the links between neighbors are established, every node sends *keepalive* messages periodically to its neighbors. Figs.28 and 29 show the procedure when the central node changes and when it does not. It is also shown in Fig.30.

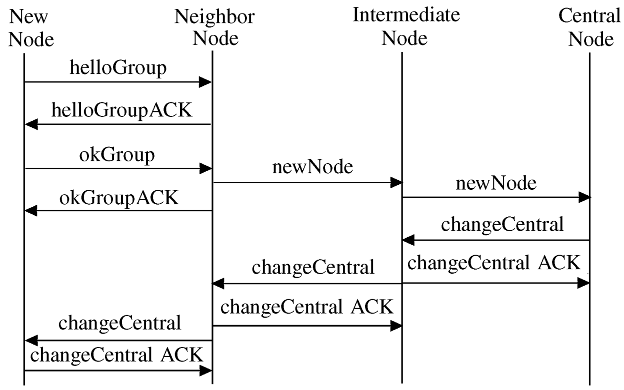


Fig.28. Messages when central node changes.

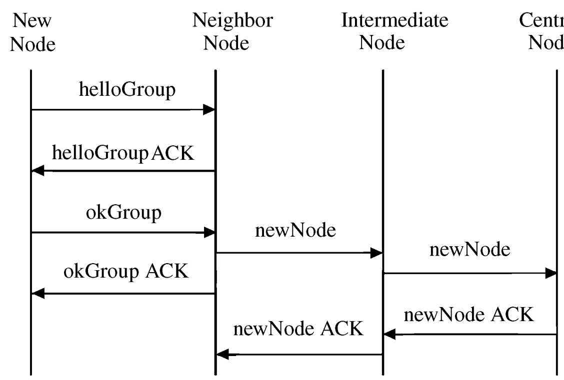


Fig.29. Messages when it does not change.

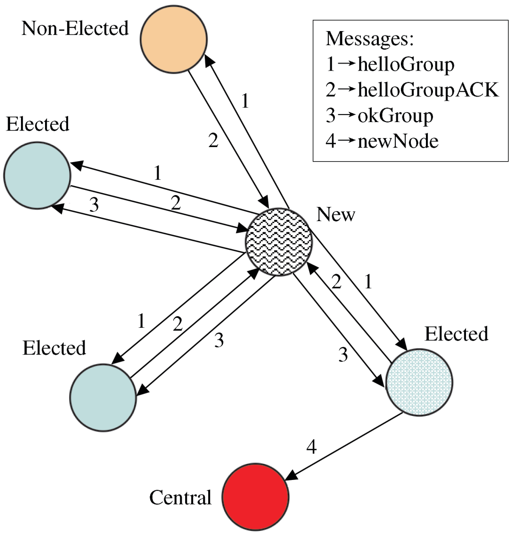


Fig.30. Message exchange when a new node joins the group.

We have proposed two choices to establish the boundaries of the group.

1) When the boundaries of the group are the same as the area that is to be covered, border nodes are known using GPS.

2) When the boundary of the group is limited by the diameter of the group, the maximum number of hops from the central node must be known. Every time a new node joins a group, it receives the *newNode ACK* message with the number of hops to the central node. When it achieves the maximum number of hops, the node is marked as a border node, and it will inform new joining nodes that they must create a new group.

6.2 Leavings and Fault Tolerance

When a node leaves the group, it will send *nodeDisconnect* message to its neighbor nodes. They must reply with a *nodeDisconnect ACK* message and send to the central node the *nodeDisconnect* message. The central node distributes the update information using RPF algorithm. If the neighbor node does not have links with other neighbors, it must start a new connection process sending a *helloGroup* message. If the leaving node is the central node, it assigns the central node role to the best candidate. This decision is taken using the value of the diameter of the group. In case of a draw, it will choose the older one in the group. Then, it sends a *changeCentral* message to the group to inform them and leaves the group. When a node fails down, its neighbor nodes will know the failure because of the absence of its *keepalive* messages. The procedure is the same as when the node leaves the network voluntarily. The central node calculates which is the best candidate, and the neighbor node will be informed by periodical *keepalive-Central* messages. New central node will distribute the update.

7 Simulations

Let T_i be the time needed by two nodes to communicate with each other, and RTT (Round Trip Time) be the mean value of the round trip time between both nodes. So, T_i can be calculated using the (8)

$$T_i = \frac{RTT_i}{2}. \tag{8}$$

The time needed to communicate a source node with a destination node in a different group is calculated using the expression given for $T_{\text{max\_intergroup}}$ in (9)

$$T_{\text{max\_intergroup}} = t_{\text{source\_border}} + \sum_{i=1}^{n} t_{\text{max\_intragroup\_}i} + \sum_{i=1}^{n+1} t_{\text{border\_}i\text{-border\_}i+1}, \tag{9}$$

n is the number of intermediate groups, $t_{\text{source\_border}}$ is the time needed to arrive from the source node to the border node in the same group, $t_{\text{max\_intragroup\_}i}$ is the time required to go through the i-th group, and $t_{\text{border\_}i\text{-border\_}i+1}$ is the time needed to transmit the information from the border node of a group to the border node of another group connected to the previous one.

We define t_p as the average propagation time for all the message transmissions between two nodes in the architecture. Its expression is shown in (10)

$$t_p = \frac{\sum_{i=1}^{m} T_i}{m}, \tag{10}$$

m represents the number of nodes involved in the path minus one. Taking into account t_p, the time needed to transmit information from the source node to the border node of the same group ($T_{\text{source\_border}}$) is defined in (11)

$$T_{\text{source\_border}} = d_{\text{source\_border}} \cdot t_p, \tag{11}$$

$d_{\text{source\_border}}$ are the number of hops needed to arrive form the source node to the border node of the same group. The maximum time to cross through a group ($T_{\text{max\_intragroup\_}i}$) is defined by the expression shown in (12)

$$T_{\text{max\_intragroup}} = d_i \cdot t_p, \tag{12}$$

i indicates the group and the d_i is the number of hops in the group. On the other hand, the number of hops for j groups is shown in (13)

$$d_i = \sum_{j=1}^{d_j} d_j. \tag{13}$$

Replacing equations in (10), (11), (12) and (13) in (9), we obtain (14)

$$T_{\text{max\_intergroup}} = \left(d_{\text{source\_border}} + \sum_{i=1}^{n} d_i + n + 1\right) \cdot t_p. \tag{14}$$

In Fig.31, we see how the interconnection time evolves between nodes of different groups.

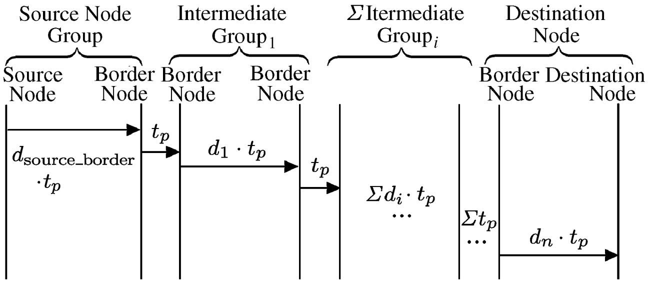


Fig.31. Connection time between nodes of different groups.

In the following subsections we are going to use (14) in order to model our proposal.

7.1 Connection Time Variation as a Function of the Number of Hops to the Border Node When All the Groups Have the Same Number of Hops

In order to do this simulation, we use a constant value for the number of intermediate groups and we varied the number of hops between the source node and the border node of its group. Then, we can observe what happens when the number of hops of the intermediate groups increases.

We have chosen the number of intermediate groups as 4. Considering that all the intermediate groups have the same number of hops, it means $d_1 = d_2 = d_3 = d_4 = d$, and introducing these values in (6) we obtain (15)

$$T_{\text{max\_intergroup}} = (d_{\text{source\_border}} + 4 \cdot d + 5) \cdot t_p. \tag{15}$$

When we give higher values to $d_{\text{source\_border}}$ for each value of d, the maximum inter group time ($T_{\text{max\_intergroup}}$) increases lineally.

7.2 Connection Time Variation When the Number of Hops to Cross the Groups Varies

This subsection studies what happens when we maintain the distance between the source node and the border node of the source group constant and we vary the number of hops of the intermediate groups and for different number of groups. We fix the parameter $d_{\text{source\_border}}$ to a value of 10. Using (14), (16) is obtained.

$$T_{\text{max\_intergroup}} = \left(11 + \sum_{i=1}^{n} d_i + n\right) \cdot t_p. \tag{16}$$

Now, we can vary d_i to observe the time needed to achieve its destination. Results are shown in Fig.32. We can deduce that the number of groups in a network does not affect the connection time to a large extent when the mean number of hops to go through the groups is small. Nevertheless, when the mean diameter of the groups is big, increasing the number of intermediate groups implies a large increase in the connection time. So, we can state that the mean diameter of the groups becomes more relevant in the calculation of the final connection time ($T_{\text{max\_intergroup}}$) for bigger networks.

In Fig.33, we can observe how the connection time varies according to the number of groups for different numbers of hops. We have chosen $d_{\text{source\_border}} = 20$, and we have varied the number of groups that will be crossed for different mean diameters of the groups, instead of varying the mean diameter of the groups.

7.3 Connection Time Variation for Different Number of Groups and Different Distances Between Source and Border Nodes in the Same Group

In this subsection we analyze how the maximum inter group time varies when we maintain the mean diameter of the group as a constant value and vary the number of groups for different distances between the source and the border nodes of the same group. To perform this experiment, we have chosen 20 as the mean diameter of the groups. (17) shows the connection time depends on the distance between the source and the border nodes in the same group and on the amount of groups in the network.

$$T_{\text{max\_intergroup}} = (d_{\text{source\_border}} + 21 \cdot n + 1) \cdot t_p. \tag{17}$$

Fig.34 shows the behavior of the $T_{\text{max\_intergroup}}$ as a function of n for several $d_{\text{source\_border}}$ values. The maximum inter group time (t_p) increases when the number of intermediate groups increases. This has happened in all the analyzed cases. Nevertheless, as we can see, there is not a big difference in the final time when we have a large or short distance between the source and the border node ($d_{\text{source\_border}}$). It means that the number of hops between the source and the border node ($d_{\text{source\_border}}$) is more relevant for having better $T_{\text{max\_intergroup}}$ that the number of groups when there are few groups. This is an important subject to take into account when designing node networks.

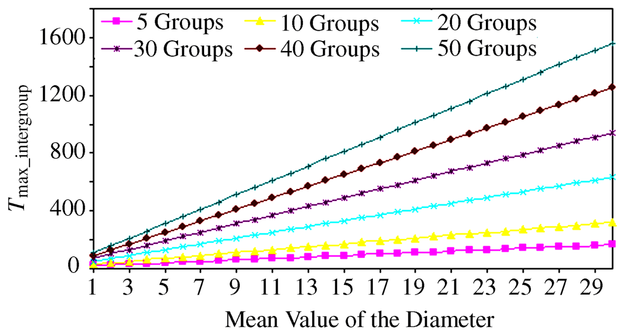


Fig.32. $T_{\text{max\_intergroup}}$ variation according to the mean diameter of the groups.

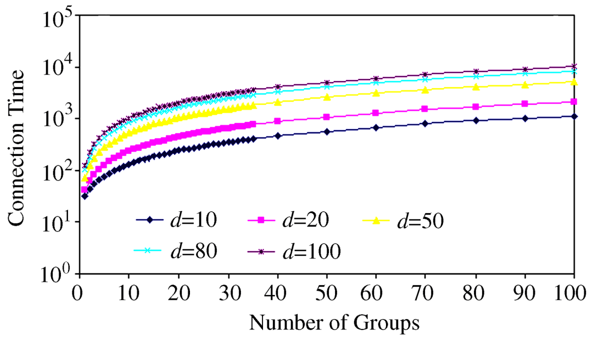


Fig.33. Connection time variation for different diameters.

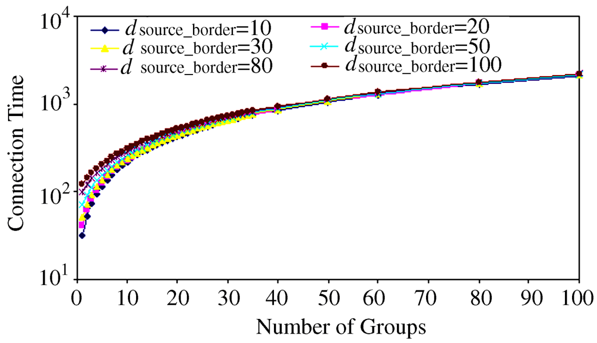


Fig.34. Connection time variation according to the number of hops to the border nodes.

7.4 Connection Time for Getting a Destination Group According to the Diameters of the Groups

In this subsection, we show the results of several simulations that give us an objective point of view about how to design a group-based node network to obtain a short connection time between two nodes belonging to different groups. We have simulated the time needed by a message sent by a node in a group until it arrives at another node of another group. Then, we observed the variation of the number of hops and the variation of the time needed to reach the destination group.

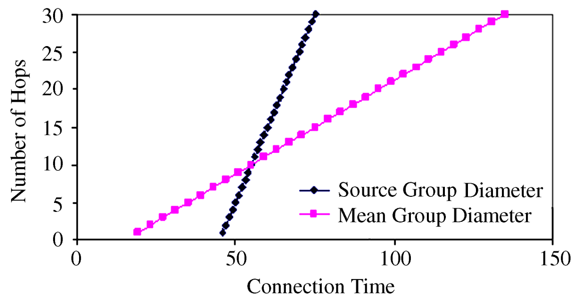


Fig.35. Connection time to reach the destination group according to the number of hops.

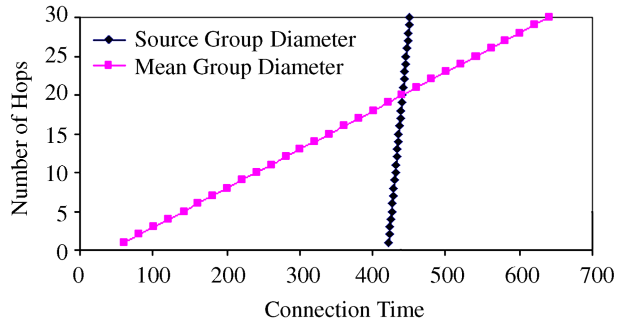


Fig.36. Connection time to reach the destination group according to the number of hops.

In Fig.35 we see the connection time of the two groups in a network with 4 groups. In order to obtain the series of the source group, we have fixed a value of 10 hops for the mean diameter of the groups and the diameter of the source group has varied between 1 and 30 hops (we have considered that groups have a maximum diameter of 30 hops). To obtain the series of the mean diameter of the group, we have fixed a value of 10 hops for the diameter of the source group and the mean diameter of the groups varies between 1 and 30 hops. As we can see, the connection time between 2 nodes increases more when the mean diameter of the intermediate groups increases. Moreover, the interconnection time between the two nodes is not so significant when the diameter of the source group increases.

In Fig.36, we have simulated the connection time between two nodes of a network with 20 groups. In order to obtain the series of the source group, a mean diameter of the groups of 20 hops has been fixed and the diameter of the source group has been varied between 1 and 30 hops. To obtain the series of the mean diameter of the group, a diameter of the source group of 20 hops has been fixed and the mean diameter of the intermediate groups varies between 1 and 30 hops.

In Figs.35 and 36, we can observe that the delay (connection time) increases when the mean diameter of the groups increases, but that increase is less significant when the number of hops from the source node to the border node of the same group increases, as we expected. Note that when we want to design a group-based network with many groups, the best solution is to increase the mean diameter of the intermediate groups instead of increasing the diameter of the source group. When a network with few groups is needed, the interconnection time varies less when we increase the number of hops in the source group.

8 Network Comparison

This section shows the comparison of our proposal with other planar group-based networks. The first one is the proposal of Xiang *et al.* (a locality-aware overlay network based on groups[31] is proposed, which has been used for Peer-to-Peer Based Multimedia Distribution Service[14]). The second one is the cluster-based network.

Table 3 shows the comparison. Our proposal stands out because of its higher efficiency in the neighbor selection system (we have added the capacity parameter), lower management cost, high fault tolerance and very high scalability.

9 Conclusions

A group-based architecture provides some benefits for the whole network. It provides fault tolerance because other groups could carry out tasks from a failed group and it is very scalable because a new group could be added to the system easily. On the other hand, a group-based network can significantly decrease the communication cost between end-hosts by ensuring that a message reaches its destination with little overheads and highly efficient forwarding. Grouping nodes increases the productivity and the performance of the network with low overheads and low extra network traffic.

In this paper we have proposed a group-based architecture where links between groups can be established by physical proximity plus the neighbor node capacity. Its operation, maintenance and fault tolerance have been detailed. Messages designed to work properly have been shown. All simulations show its viability and how it could be designed to improve its performance. Finally we have compared it with another group-based logical architecture to show their differences.

Table 3. Planar Group-Based Topologies Comparison

	Locality-Aware Overlay Network (Z. Xiang *et al.*)	Cluster Based Topologies	Our Group-Based Proposal
Need of a Rendezvous Point	Yes	No	No
Nodes with Higher Role	No	Yes	No
Type of Topology	Logical (but it could be implemented in physical)	Physical and Logical	Physical (but it could be implemented in logical)
Neighbor Selection	Proximity in the Underlying Network (IP)	Physical Proximity	Physical Proximity + Capacity
Which Group to Join In	Based on Rendezvous Point Decision + Boot Nodes	Proximity	Based on Neighbor Discovery (time to reply or closest)
Management Cost	Medium (because of the rendezvous point)	Medium (because of cluster head)	Low
Fault Tolerance	Very Low (because Rendezvous Point or boot nodes failure)	Low (because cluster head failure)	Very Much
Scalability	Very Much (depending on the RP)	Medium	Very Much
Availability	Low (when boot nodes from head a group are not available, the group is not available)	Low (when a cluster head is not available, the group is not available)	Very High (when a sensor finds a neighbor it joins the network)

The architecture proposed can be used for specific cases or environments, such as the ones which require the set up of a network where groups appear and join the network or by networks that are wanted to be split into smaller zones to support a large number of sensors. There are many application areas for this proposal such as rural and agricultural environments or even for military purposes. Now, we are programming the protocol for a specific wireless sensor device to test it over a real environment.

References

[1] Akyildiz I F, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. *IEEE Communications Magazine*, 2002, 40: 102–114.

[2] Frodigh M, Johansson P, Larsson P. Wireless ad hoc networking. The art of networking without a network. *Ericsson Review,* 2000, (4): 248–263.

[3] Kumar S, Raghavan V S, Deng J. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks,* May 2006, 4(3): 326–358.

[4] Royer E M, Toh C K. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications,* April 1999, 6(2): 46–55.

[5] Rajaraman R. Topology control and routing in ad hoc networks: A survey. *ACM SIGACT News,* June 2002, 33(2): 60–73.

[6] Ratnasamy S, Handley M, Karp R, Shenker S. Topologically-aware overlay construction and server selection. In *Proc. InfoCom*, New York, USA, June 2002, pp.1190–1199.

[7] Lv Q, Ratnasamy S, Shenker S. Can heterogeneity make Gnutella scalable? In *Proc. the First International Workshop on Peer-to-Peer Systems*, MIT, Cambridge, MA, USA, March 7∼8, 2002, pp.94–103.

[8] Woolston K, Albin S. The design of centralized networks with reliability and availability constraints. *Computers and Operations Research,* May 1988, 15(3): 207–217.

[9] Liu-Sheng Huang, Hong-Li Xu, Yang Wang, Jun-Min Wu, Hong Li. Coverage and exposure paths in wireless sensor networks. *Journal of Computer Science and Technology,* July 2006, 21(4): 490–495.

[10] Jaeook Lee, Sun Kang. Satellite over satellite (SOS) network: A novel architecture for satellite network. In *Proc. IEEE Infocom 2000,* Tel Aviv, Israel, March 26∼30, 2000, pp.315–351.

[11] Ganz A, Krishna C M, Tang D, Haas Z J. On optimal design of multitier wireless cellular systems. *Communications Magazine, IEEE,* February 1997, 35(2): 88–93.

[12] Chong-Yi Yuan, Wen Zhao, Shi-Kun Zhang, Yu Huang. A three-layer model for business processes — Process logic, case semantics and workflow management. *Journal of Computer Science and Technology,* May 2006, 22(3): 410–425.

[13] Wierzbicki A, Strzelecki R, Swierczewski D, Znojek M. Rhubarb: A tool for developing scalable and secure peer-to-peer applications. In *Proc. Second IEEE International Conference on Peer-to-Peer Computing,* Linköping, Sweden, 5∼7 September 2002, pp.144–151.

[14] Xiang Z, Zhang Q, Zhu W, Zhang Z, Zhang Y. Peer-to-peer based multimedia distribution service. *IEEE Transactions on Multimedia,* April 2004, 6(2): 343–355.

[15] Hongjun L, Luo L P, Zhifeng Z. A structured hierarchical P2P model based on a rigorous binary tree code algorithm. *Future Generation Computer Systems,* February 2007, 23(2): 201–208.

[16] Thallner B, Moser H. Topology control for fault-tolerant communication in highly dynamic wireless networks. In *Proc. the 3rd International Workshop on Intelligent Solutions in Embedded Systems*, Hamburg University of Technology, Hamburg, Germany, May 20, 2005, pp.89–100.

[17] Yu J Y, Chong P H J. A survey of clustering schemes for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials,* 2005, 7(1): 32–48.

[18] Jiang M, Li J, Tay Y C. Cluster based routing protocol (CBRP). Internet-draft, draft-ietf-manet-cbrp-spec-01.txt, National University of Singapore, August 14, 1999.

[19] Chiang C C, Wu H K, Liu W, Gerla M. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel. In *Proc. the IEEE SICON'97*, National University of Singapore, Kent Ridge, Singapore, April 14∼17, 1997, pp.197–211.

[20] Zygmunt J Hass, Marc R Pearlman, Prince Samar. The zone routing protocol (ZRP) for ad hoc networks. IETF, Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

[21] Haas Z J, Pearlman M R. The Zone Routing Protocol: A Hybrid Framework for Routing in Ad Hoc Networks. Ad Hoc Networks, Perkins C E (ed.), Reading, MA: Addison-Wesley, 2000.

[22] Mirco Musolesi, Cecilia Mascolo. A community based mobility model for ad hoc network research. In *Proc. Second International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality, REALMAN 2006*, Florence, Italy, May 26, 2006, pp.31–38.

[23] Tsuchiya P F. The landmark hierarchy: A new hierarchy for routing in very large networks. *Computer Communication Review,* August 1988, 18(4): 35–42.

[24] Rekhter T, Li T. A border gateway protocol 4 (BGP-4). RFC 1771, March 1995.

[25] Pei G, Gerla M, Hong X. LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. In *Proc. IEEE/ACM MobiHOC*, Boston, MA, USA, August 2000, pp.11–18. .

[26] Srdjan Krco. Health care sensor networks — Architecture and protocols. *Ad Hoc & Sensor Wireless Networks,* 2005, 1(1/2): 1–25.

[27] OPNET Modeler website. http://www.opnet.com/solutions/network_rd/modeler.html.

[28] P Gober, A Ziviani, P Todorova *et al.* Topology control and localization in wireless ad hoc and sensor networks. *Ad Hoc & Sensor Wireless Networks,* Oct. 2005, 1(4): 301–321.

[29] McQuillan J M, Richer I, Rosen E C. The new routing algorithm for the ARPANET. *IEEE Transactions on Communications,* May 1980, 28(5): 711–719.

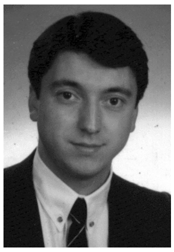
[30] Dalal Y K, Metcalfe R M. Reverse path forwarding of broadcast packets. *Communications of the ACM,* December 1978, 21(12): 1040–1048.

[31] Zhang X, Zhang Q, Zhang Z, Song G, Zhu W. A construction of locality-aware overlay network: mOverlay and its performance. *IEEE Journal on Selected Areas in Communications,* January 2004, 22(1): 18–28.

Jaime Lloret received his M.Sc. degree in physics in 1997 from University of Valencia and a postgraduate Master in corporative networks and systems integration in 1999. He received his M.Sc. degree in electronic engineering in 2003 from University of Valencia and his Ph.D. degree in telecommunication engineering from the Polytechnic University of Valencia in 2006. He obtained the first place given by the Spanish Agency for Quality Assessment and Accreditation for the Campus of Excellence in the New Technologies and Applied Sciences area and he was awarded the prize of the best doctoral student in the telecommunications area in 2006 according to the Social Council of the Polytechnic University of Valencia. He is a Cisco Certified Network Professional Instructor. He worked as a network designer and administrator in several companies. He has been editor of several conference proceedings and associated editor and guest editor of several international journals. Dr. Lloret has been involved in more than 40 program committees of international conferences till 2008. He was the chairman of SENSORCOMM 2007 and UBICOMM 2008.

Miguel Garcia received the M.Sc. degree in telecommunication engineering in 2007 from Polythecnic University of Valencia in Spain. He is currently a Ph.D. candidate in the Department of Communications of the Polythecnic University of Valencia. His major research interest includes sensor, ad-hoc and P2P networks, and grid computing. He has been technical committee member in several conferences. He is a CCNA Cisco instructor and an IEEE student member.

Jesus Tomás graduated in computer science in 1993 from Polytechnic University of Valencia. He finished his dissertation in 2004. From 1993 he is a lecturer at Polytechnic University of Valencia. He is member of the Pattern Recognition and Human Language Technology Group, the "Instituto Tecnológico de Informática". His main research focus is on speech recognition and machine translation. In these areas, he has been involved in a number of projects with public and private budgets.



Fernando Boronat was born in Gandia, Spain and went to the Polytechnic University of Valencia (UPV) in Spain, where he obtained, in 1993, his M.Sc. degree in telecommunications engineering. From 1994 he worked for a couple of years for Telecommunication Companies. In 1996 he moved back to the UPV, where he is a lecturer in the Communications Department at the Escuela Politécnica Superior de Gandia. He obtained his Ph.D. degree in 2004 and his topics of interest are communication networks, multimedia systems and multimedia synchronization protocols. He is a member of IEEE since 1993 and is involved in several TPCs of national and international conferences.

A Group-based Content Delivery Network

Jaime Lloret[1], Miguel Garcia[2], Diana Bri[3], Juan R. Diaz[4]
Communications Department
University Polytechnic of Valencia
Camino Vera s/n, 46022, Valencia, Spain
[1]jlloret@dcom.upv.es, [2]migarpi@posgrado.upv.es, [3]diabrmo@posgrado.upv.es, [4]juadasan@dcom.upv.es

ABSTRACT
It is known that a group-based system provides better performance and more scalability to the whole system while decreasing the communication traffic. Group-based architectures in Content Delivery Networks (CDNs) could be a good solution to the need of scalability. There is not any group-based CDN in existence, although we proposed in reference [1] a group-based system to interconnect CDNs of different providers. This article shows a Content Delivery Network based on grouping surrogates. We will show the benefits of our proposal and its application environment. We will describe the protocol developed to connect surrogates from the same group and from different groups. The neighbor selection algorithm is based on their proximity in order to provide lower content distribution times and trying to assure Quality of Service (QoS) by connecting to surrogates with higher available capacity. Real measurements of the network control traffic and of the performance of the surrogates will be shown. Finally we will show the differences with the system proposed in reference [1].

Categories and Subject Descriptors
C.2.1 [Network Architecture and Design] Network communications; Network topology

General Terms
Algorithms, Management, Measurement, Performance, Design.

Keywords
Group-based network, group-based architecture, CDNs.

1. INTRODUCTION AND RELATED WORKS
The idea behind CDNs consists on placing separate servers, called surrogates, near to the client location. If the user is redirected to a nearby surrogate, which acts as a proxy, it could experience a significant reduction in the perceived response time. A CDN acts as a trusted overlay network that offer high-performance delivery of common web objects, static data, and rich multimedia content by distributing content load among servers that are close to the clients. A Content Delivery Network provides scalability, fault tolerance, and load balancing for the delivery of content and media streaming. CDNs were developed to minimize the latency to deliver the content to the clients and to overcome performance problems, such as network congestion and server overload, that arise when many users access popular content. Since content is delivered from the closest edge server and not from the origin source, the content is sent over a shorter network path, thus reducing the request response time, the probability of packet loss, and the total network resource usage. While CDNs were originally intended for static web content, recently, they have been applied for delivering media streaming as well [2].

The communication process inside a CDN can be divided into two separate networks:

1. The distribution network, between origin site and surrogates.
2. The delivery network, between surrogates and clients.

The collection of surrogates that compound the CDN replicate content of the origin server [3]. We can find several deployed CDNs that deliver many types of content over Internet. Akamai [4], which acquired Speedera, Nine Systems, Digital Island and others, provides a distributed computing platform that delivers web, media streaming, software and applications. Another popular CDN is CODIS [5], which aim is to deliver documents and media streams over a satellite-based CDN using a central high-bandwidth satellite as a single-hop backbone for a continental CDN. We can also find mobile CDNs such as MarconiNet [6], an IP-based radio and TV network built on standard Internet protocols. Moreover, there are commercial products developed by different vendors, such as Cisco's ECDN solution, that is being used for e-learning and for IP/TV broadcasting, or Nortel's Content Director/Cache [7], that is being used to deliver media streaming. There are also general purpose open developments of CDNs such as Globule [8], Coral [9], CoDeeN [10] with different structure and operation protocols. A Model for Content Internetworking is published in the RFC 3466 [11]. Although it does not explain how to interconnect the network components, it explains the different components that should have a CDN. There are several works that study the interconnection system between surrogates. An example is given in reference [12], which gives three models to develop CDNs: based on a P2P system, on a GRID system or on an agent-based system. The fact of developing a new model means having a new interconnection system among the CDN surrogates.

The overlay network with a flat topology does not scale well [13]. Hierarchical overlay network topology is required for a large-scale Content Delivery Network to perform content delivery

UPGRADE-CN'08, June 23, 2008, Boston, Massachusetts, USA.

scalable and efficiently. Grouping nodes into clusters is one of the schemes most used when scalability is needed [14]. Several schemes can be used to organize the surrogates, through manual configuration or through a self-organizing scheme [15]. The objective of this work is to develop a Content Delivery network where surrogates are self-organized into groups taking into account their position and where the surrogates in group will have connections with surrogates of neighboring groups. These connections are established only if the surrogates have a distance lower than a predefined value and on the Round Trip Time (RTT), and in case of several choices from the same group the election is taken based on several parameters that will be explained in a later section.

In [1], we presented a new interconnection system between several content delivery servers without varying the initial model of the CDN. It was deployed to be applied over any CDN in existence or to interconnect existing CDNs and with the purpose of joining them. The connections between surrogates are based on the available capacity of the servers. We will discuss the difference between the one presented in this paper and the one presented in reference [1].

The remainder of this paper is organized as follows. Section 2 shows the group-based networks benefits and where a group-based CDN could be applied. Section 3 gives the architecture description. Joining, leavings, fault tolerance and architecture operation are described in section 4. Section 5 shows real measurements of the group-based CDN operation and of its surrogate performance. Section 6 concludes the paper and gives our future works.

2. GROUP-BASED SYSTEM BENEFITS AND APPLICATION ENVIRONMENTS

A group is referred as a small number of interdependent nodes with complementary operations that interact in order to share resources or computation time, or to acquire content or data and produce joint results. In a physical group-based architecture nodes are close (in terms of geographical location or RTT) to each other. In this paper we present a new CDN architecture which brokers are structured in groups and their connections are established regarding their physical proximity (although it could be changed by GPS positions or IP addresses) and their available capacity.

Generally, group-based systems have been designed to solve specific issues. In the literature we can find some of them. One of them is the Rhubarb system [16], which organizes nodes in a virtual network, allowing connections across firewalls/NAT. Another one is the Peer-to-Peer Based Multimedia Distribution Service presented in [17], where a topology-aware overlay, in which nearby hosts or peers self-organize into application groups, is proposed. And there are some hierarchical architectures were nodes are structured hierarchically and parts of the tree are grouped into groups [18] [19].

Group-based networks provide some benefits for the whole network such as:

- Spreads the work to the network in groups giving more flexibly, efficiently and lower delays.
- Content availability will increase because it could be replicated to other groups.
- Any surrogate could receive content from every group using only one service.
- It provides fault tolerance. Other groups could carry out tasks from a failed one.
- Network measurements could be taken from any group.
- It is more scalable because new surrogates and new groups could be easily added to the system.

A group-based network allow the interaction between content delivery groups and, by spreading work to the network, give the capability to operate more flexibly, efficiently and less time consuming without the delays and information congestion of a strict workflow system. There are some works in the literature that shows the benefits of group-based schemes [20].

On the other hand, a group-based network can significantly decrease the communication cost between end-hosts by ensuring that a message reaches its destination with small overhead and highly efficient forwarding. So, grouping nodes increases the productivity and the performance of the network with low overhead and low extra network traffic. Therefore, good scalability can be achieved in group-based architectures.

There are many application environments where a group-based topology can be applied. Some of these cases are the following:

1. Let's suppose a CDN where the users of a geographical zone use to receive a specific content different from other geographical zones because of cultural issues (although content from other zones have to be available), if we split the CDN into groups, the performance of the CDN will be increased.
2. Let's suppose a CDN that delivers different types of content, and surrogates have to be grouped taking the content in mind to provide lower delays between them or to provide higher QoS.
3. Let's suppose a Wireless CDN. Surrogates are connected because of the surrogate's coverage area, so physical connectivity is the main issue and a group-based topology based on surrogate's proximity could be the best deployment.

3. ARCHITECTURE DESCRIPTION

From the logical point of view, the architecture is based on a two-layer model. Surrogates in the upper layer are called control surrogates (CS). They control the group and any surrogate has to establish a connection with it to join its group. In our design we have provided only one CS per group, but there could be added more for scalability purposes. CSs have connections with some CSs of others groups and with all lower layer surrogates, also called Distribution Surrogates (DS), in its group. The information between CSs is routed using SPF algorithm [21], but it could be changed by any other routing algorithm. The way the SPF algorithm could be applied to this network can be seen in reference [22]. CSs are used to organize connections between DSs of different groups. Any group must have a CS that must have connections with other CSs of the CDN. DSs give service to final users. DSs have connections with elected DSs of other groups in order to provide the content that is being distributed in the other groups. CS is also a DS. All groups have a CS and one or several DSs, so all groups must have both layers.

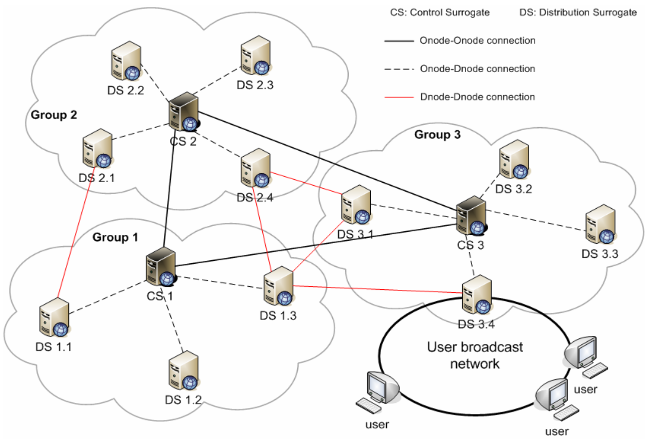


Figure 1. Proposed architecture topology example.

An example of the architecture proposed is shown in figure 1. CSs have connections with some CSs of other groups (solid black lines). DSs have a connection with the CS of its group (lines formed by black points) and with the selected DCs of the other groups (solid red lines).

In order to assure enough process capacity in the CS and DS to perform their tasks, we have defined several limitation parameters:

- CS_max_con: It is the maximum number of connections that a CS can have with other CSs.
- DS_max_con: It is the maximum number of connections that a DS can have with other DSs.
- Max_load: It is the surrogate maximum load due to their content distribution workload plus the CDN management.
- Max_distance: It gives the maximum distance in which CSs and DSs are able to establish connections. Although we will use this parameter as it were in meters or kilometres, it could be used as global position (GPS) differences or virtual proximities given by IP distances.
- Max_RTT: It gives the maximum RTT value permitted to establish a connection. Surrogates that give higher RTT values will not be taken into account to establish a connection.

We have defined λ as the capacity of a surrogate. It depends on the surrogate's upstream and downstream bandwidth (in Kbps), its number of available connections (Available_Con) and its % of available load. It is used as one of the parameters to determine the best surrogate to connect with. In order to define surrogate's bandwidth weight in the calculation of λ, surrogates with total bandwidth (upstream plus downstream) equal or lower than 256 Kbps have the same weight. λ parameter is defined by equation 1.

$$\lambda = \text{int}\left[\frac{BW_{up} + BW_{down}}{256} + 1\right] \cdot Available\_Con \cdot (100 - load) \qquad (1)$$

There are two types of λ, one for CSs and another for DSs. If Available_Con (available number of connections) is higher than CS_max_con or DS_max_con, depending on the case or load (load of the surrogate) is higher than Max_load, λ is equal to 0.

All connections between CSs, between DSs or between a DS and a CS are established taking into account the distance between them, the RTT between them and the λ of the other surrogate. If the distance is higher than Max_distance or RTT is higher than Max_RTT, this new connection is rejected.

CSs will have a table with the CSs of other groups and with the DSs in its group. All DSs will have an entry with the CS of its group and a table with the elected DSs of other groups. All these tables will have the distance, the RTT and λ parameter for each entry.

4. PROTOCOL AND ARCHITECTURE OPERATION

When a new surrogate joins the CDN must have configured CS_max_con, DS_max_con, Max_load BWup and BWdown and its position (it could be given manually, by GPS or using the IP address).

When a surrogate appears in the network, first, it sends a "discovery" message with its position to surrogates previously known manually or by Bootstrapping [23]. If it does not receive any response it becomes a CS, so it creates a group. All CSs that receive this discovery message will reply with a "discovery ack" message which has its IP, its λ parameter and the relative distance. It will wait replies for 10 seconds. If it receives replies from DSs or from CSs, but the CSs have a distance higher than Max_distance, or a RTT higher than Max_RTT (obtained measuring the response delay) or λ=0, it becomes a CS.

Once it is a CS, it sends a "C connect" message to establish connections with selected CSs (based on the combination of the distance, the RTT and the λ parameter). If the other CS agrees that connection, it adds this entry to its CS table and sends a "Welcome C" message with all IDs of the groups in the CDN. The CS will create randomly a groupID between available values. Then, the new CS will send "keepalive C" messages periodically with its groupID to all its neighbors from other groups to indicate it is alive. If the new CS does not receive a keepalive message from the CS for a deadline time, it would erase that entry from the database. Steps followed by the protocol when a new surrogate arrives to the CDN and becomes a CS are shown in figure 2.

CSs are used to manage the architecture. They allow the establishment of connections between DSs of different groups. In order to broadcast any information to all the CSs in the CDN we have used the Reverse Path Forwarding algorithm [24], which routes the packets based on the SPF tree.

Otherwise, if the surrogate receives a "discovery ack" reply from a CS which distance is equal or lower than Max_distance, or a RTT lower than Max_RTT or λ=0, the new surrogate will choose the best CS to have a connection with (taking in mind these parameters) and becomes a DS. The weigh of the distance, RTT and λ parameter in the election is given by the network designer decision. An example will be given in the test bench used to take measurements. Then, the new DS will send a "D connect" message to the CS in the group. It will add that entry to its DS table. Finally, it will send to the CS keepalive messages periodically to indicate that it is still alive. If the CS does not receive a keepalive message from the DS for a dead time, it will erase this entry from its database. Figure 3 shows steps explained.

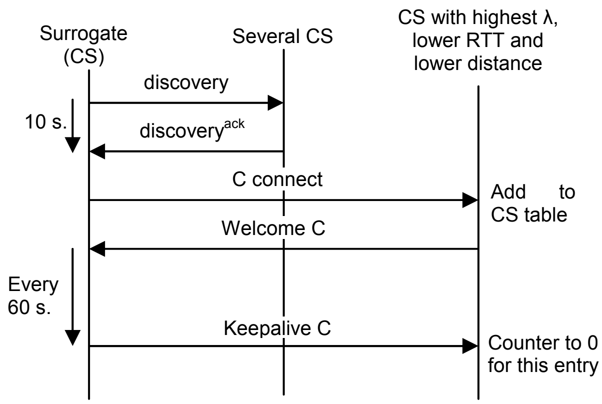


Figure 2. Protocol operation when there is a new CS.

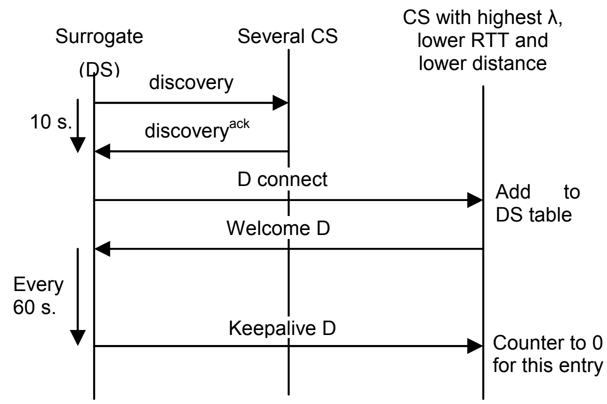


Figure 3. Protocol operation when there is a new DS.

When there is a new DS, it has to establish connections with DSs from other groups. First, it has to send a "DDB request" message to the CS in its group. This message has the requester IP and is routed through the CS's network using the groupID. When a CS receives that message, it chooses the best DS using the position and the λ parameter of the DSs in the DS table. Then, the CS sends the "elected DS" message to it and waits for an "elected DS ack" message to be advised that the DS knows this election. Next, the elected DS will contact the new DS directly using a "DD connect" message.

When the new DS receives the "DD connect" message, it adds this entry to its DS-DS table and will reply with a "Welcome DD" message. Then, the second DS will add this entry to its DS-DS table. Finally, both will send keepalive messages periodically to indicate that they are still alive. If anyone of them doesn't receive a keepalive message for a dead time, it will erase this entry from its database, so it will send a "DD request" for this group. Steps explained are shown in figure 4.

When a DS leaves the CDN voluntarily, it will send a "D disconnect" message to the CS node of its group. The CS node will erase this entry from its DS table. And it will also send a "DD disconnect" message to the DSs from other groups which has connections. They will erase this entry from their DS-DS table and will look for a new DS for this group if they don't have. If the node fails down, because CS and DSs from other groups send keepalive messages, they will know that it has leaved the CDN and they will erase that entry from their tables.

The nearest DS of CS will be the backup CS. The backup CS will have the same information of the CS. The CS sends keepalive messages periodically to the backup CS. When a CS leaves the CDN voluntarily, it has to send a "CS disconnect" to its neighbor CSs and they will erase that entry from their CS table. That update will be propagated through the CS network using the Reverse Path Forwarding algorithm. The leaving node will send a "CS disconnect" to the backup CS in its group. Then, it will leave the CDN. Because the backup CS has the DS table of its group and the CS table of the failed CS, it will become CS and will send a "C connect" message to establish connections with CSs of other groups. Figure 5 shows the described procedure.

If the CS fails down, it will be known because of missing keepalives. Both neighbouring CSs and the backup CS will notice it. If the backup CS does not receive a keepalive from the CS for a dead time, it will become a CS and it will proceed in the same manner that when the CS leaves the CDN voluntarily.

5. SYSTEM MEASUREMENTS

This section shows the real measurements taken from the control messages and the performance of the surrogates of a deployed CDN.

5.1 Testbench

We have developed a desktop application tool, using Java programming, to run and test the designed protocol and the CDN performance. Object-oriented programming allows us to have several modules, so we can change easily parts of the application to adapt it to different types of CDNs. We have programmed CS and DS functionalities. The application allows us to configure some parameters such as the maximum number of connections, maximum load, upstream and downstream bandwidth, keepalive time and so on). The application calculates the λ parameter internally.

The test bench was formed by 14 Intel ® Celeron computers (2 GHz, 256 MB RAM) with Windows 2000 Professional Operative System. They were connected to a Cisco Catalyst 2950T-24 Switch over 100BaseT links. One port was configured in a monitor mode (receives the same frames as all other ports) to be able to capture data using a sniffer application.

In order to know the control traffic of the developed group-based Content Delivery Network, we have placed 14 subrogates in the position shown in table 1. The values of the parameters are shown in table 2. Then, we have started the nodes by sequentially order every 30 seconds. The topology obtained for the test bench and the physical position of the surrogates is shown in figure 6. It also shows the connections between CSs (solid black lines), between DSs and the CS in their group (solid red lines) and between DSs (lines formed by black points). We have implemented the position only using 2 dimensions, but it could be modified for 3 dimensions.

5.2 Network real measurements

In order to see the performance of the developed CDN, we measured their behavior in the phase of initialization. That is, we observed how performs the CDN as the surrogates join the network taking into account their position, the distance with other nodes and the RTT.

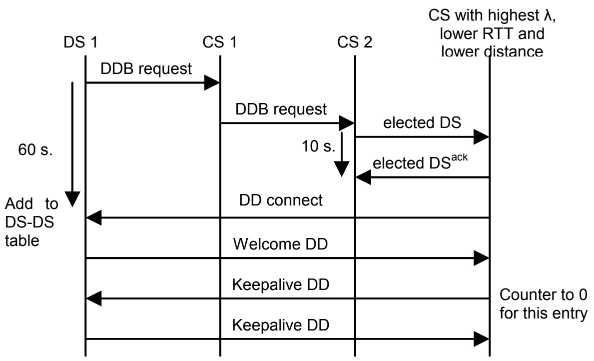


Figure 4. Protocol operation to establish a connection with a DS of other group.

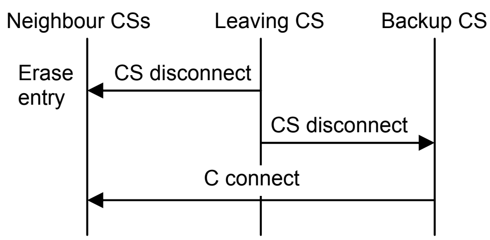


Figure 5. Protocol operation when a CS leaves the CDN.

Table 2. The values of the parameters used to take measurements.

Parameters	**Values 1st case**
Upstream Bandwidth	1024 Kbps
Downstream Bandwidth	256 kbps
Keepalive Time	30 seconds
Holdtime	60 seconds
Maximum CPU utilization	100 %
CS_max_con	4
DS_max_con	4
Max_distance	1000
Max_RTT	50 mseconds

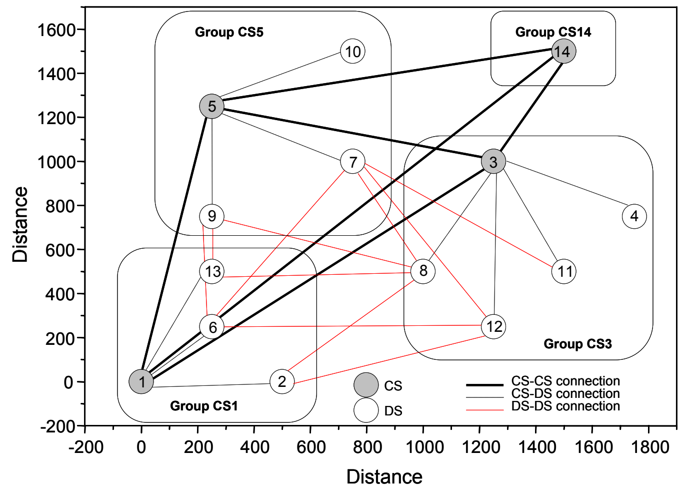


Figure 6. Surrogates distribution and connections established when the CDN has been set up.

Table 1. Surrogate's position.

Surrogate	X position	Y position
1	0	0
2	500	0
3	1250	1000
4	1750	750
5	250	1250
6	250	250
7	750	1000
8	1000	500
9	250	750
10	750	1500
11	1500	500
12	1250	250
13	250	500
14	1500	1500

We observed that when all these nodes started sequentially, only 4 groups were created and the CSs were surrogates 1, 3, 5 and 14.

The number of broadcasts sent by the nodes when the CDN is setting up is observed in figure 7. We observed that there are peaks of broadcast due to new joining nodes. An important feature is that the number of broadcasts in the network is equal to the number of existing groups at that time. Figure 8 shows the amount of control traffic introduced by our CDN. We can see that it uses only 1024 Bytes/s the entire network when there are 14 surrogates. There are peaks each 60 seconds approximately because of keepalive messages and joining nodes in the initial process. When the network has converged, the control traffic is very similar to the stage of initialization. It happens because the developed architecture does not provide a high control traffic load in the initial phase. It allows us to demonstrate that it will introduce very little additional traffic when the topology changes.

When we measured the number of packets in the network (figure 9), we got 16 packets per second when there were not any entry in the network. There are peaks of around to 34 packets per second when there are new joining nodes added to keepalive messages.

When the network converges, we got the same number of packets per second.

Figure 10 shows the behaviour of the CDN when the network has converged. We have observed that the number of broadcast decreased. There are peaks when DSs try to find DSs of other groups. At 120 seconds, we noted that there were many broadcasts. It was because we introduced a new node in the network.

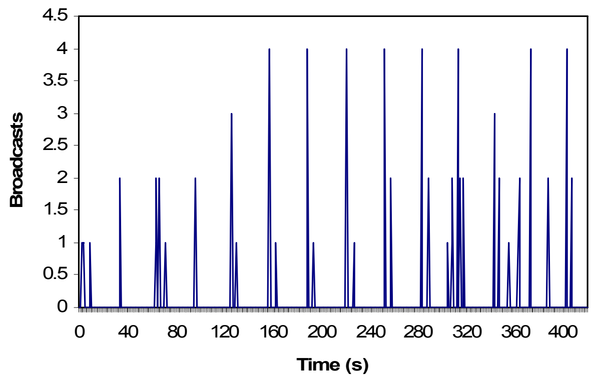


Figure 7. Broadcasts in initial process.

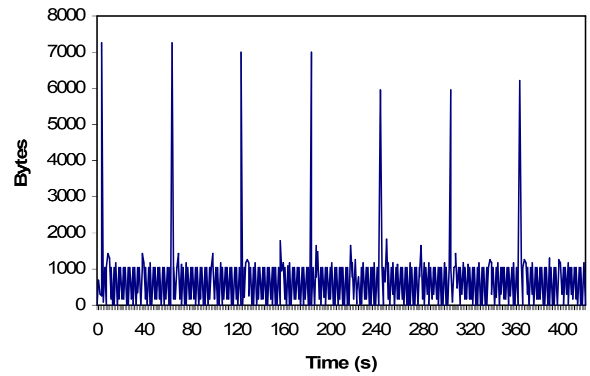


Figure 8. Control traffic in initial process (bytes/s).

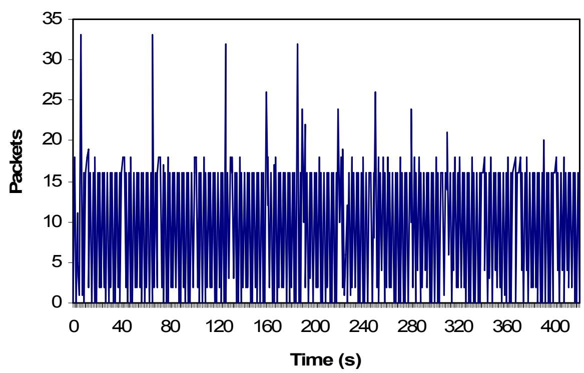


Figure 9. Control traffic in initial process (packets/s).

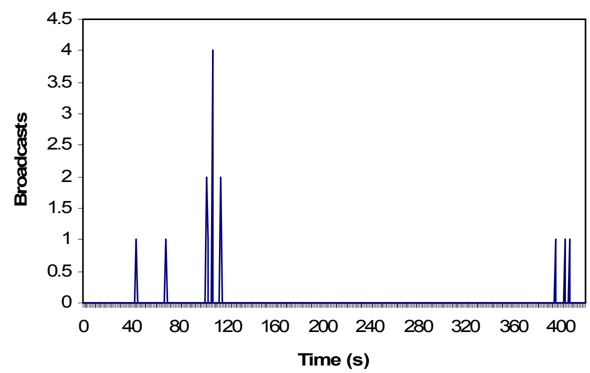


Figure 10. Broadcasts when the CDN has converged.

5.3 Surrogate real measurements

In order to obtain the surrogate measurements, we considered three scenarios. The first scenario was given by surrogate 11. It was distributing video to surrogate 7 while it was sending video streams to its user's network. The second scenario was given by surrogate 9. It was receiving video from surrogate 13 while it was sending video streams to its user's network. The third scenario was given by surrogate 2. It was distributing video to surrogates 8 and 12 while it was sending video streams to its user's network. The following graphs summarize all our measurements and results.

Figure 11 shows the % of processing time measured in surrogate 11 when it distributed 200 MB while it was sending video streams to its user's network at the same time. We have observed that the surrogate requires more processing time when it sends video content to another surrogate (an average of 53.62% of processing time with a standard deviation of 6.33%) than when it sends video streams to the user's network (it uses an average of 32.82% of processing time with a standard deviation of 4.55%). The difference between them has been 20.8%). This variation is due to a computer, by default, needs more processing time to read a file and send it to another surrogate than to send streams to the final users.

In order to observe the behavior of the network, we have incremented the size of the content distributed in the CDN in the second scenario. Figure 12 shows the % of processing time measured when it is being distributed 800 MB. The time needed to transmit the file is greater than the case of the figure 11. We expected 4 times more because the content size was 4 times more, but our surprise was that the time to transmit this content was reduced 37% of the expected time. The processing time had an average of 53.61 % when the content is distributed to other subrogates and 32.82 % when the surrogate sends video streams to its user network. The % of processing time is higher when it is receiving than when it is sending.

In figure 13 we have compared the number of readings of the cache, when the content transmitted has a size of 1 GB, of surrogates 11, 9 and 2. Surrogate 11 gives 91 readings of cache per second in average. Surrogate 9 gives around 6 readings of cache per second. We think that it was because surrogate 9 was receiving content instead of sending it, therefore, the number of readings in cache is very low. The worst case was when the surrogate was distributing content in parallel to two subrogates while it is sending video streams to its user's network. In this case, the number of readings in cache per second is around 330.

6. CONCLUSIONS

We have shown the development of a two layer CDN architecture that allows grouping surrogates and establishes connections between surrogates of neighboring groups taking into account their distance and a parameter which is based on the capacity of the surrogate. Control Surrogates manage the CDN and Distributed Surrogates allow interconnections between groups of the CDN. We have described the protocol developed and the flow of the designed messages. The CDN can be easily deployed over IPv6 because its information is transmitted using group identifiers.

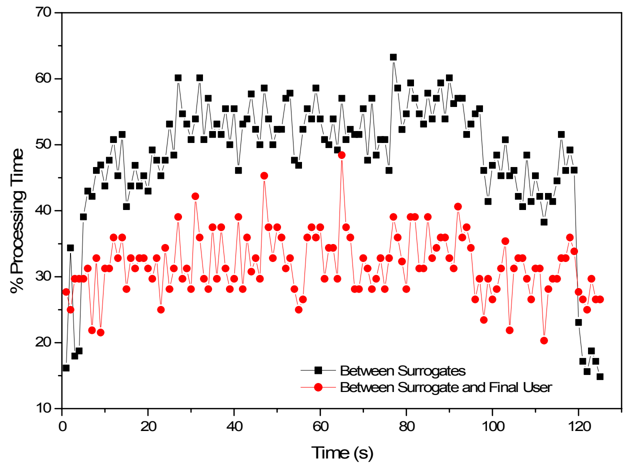


Figure 11. % processing time when it is distributed 200 MB.

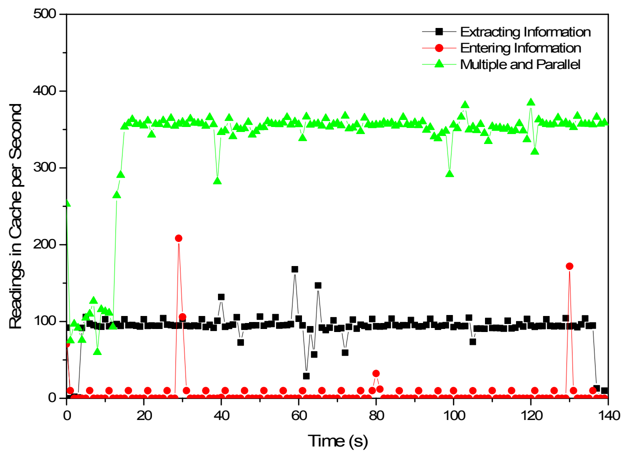


Figure 13. Readings in cache per second when it is distributed 1 GB.

We have shown the performance of the network and how surrogates perform in different execution cases. It has demonstrated that the CDN requires low bandwidth to run and work properly. The case of study used to take measurements has shown how the CDN is set up from the beginning.

In reference [1] we presented a group-based system to establish connections between content delivery server groups were groups have their own protocol, so the system could be applied over any CDN in existence or to interconnect different CDNs. There, connections between surrogates were based on the capacity of the surrogate and there were a promotion parameter. In this case, we have presented a new CDN which surrogates are grouped taking into account many parameters such as their proximity, the RTT and the capacity of the surrogates to provide QoS. In addition, connections between surrogates of different groups are established taking into account all those parameters and, on the other hand, there is not any promotion parameter such in the one presented in reference [1]. We can state that both architectures are very different, not only because of the type of parameters used, but of the CDN structure and operation.

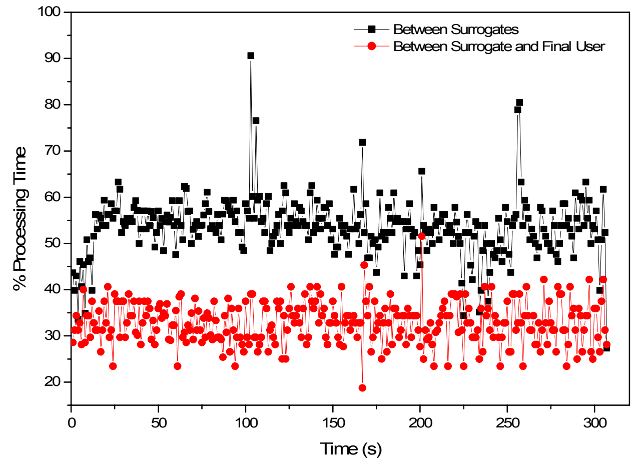


Figure 12. % processing time when it is distributed 800 MB.

Now, we are introducing in our scheme the possibility of changing the connections dynamically based on the λ parameter or on the type of content that is being transferred. We will also make a variation in our basic scheme where connections between surrogates will be made as a function of the type of content that is being transferred to enhance the performance of the network.

7. REFERENCES

[1] Jaime Lloret, Carlos Palau and Manuel Esteve. 2008 Structuring connections between content delivery servers groups. Future Generation Computer Systems, Elsevier Editorial, Vol. 24, Issue 3, Pages 191-201.

[2] John Apostolopoulos, Tina Wong, Wai-tian Tan, Susie Wee. 2002 On multiple decription streaming with content delivery networks, IEEE Infocom.

[3] A. Vakali, G. Pallis. 2003 Content delivery networks: status and trends, IEEE Internet Computing 7 (6), pp. 68-74.

[4] Akamai, available from: http://www.akamai.com

[5] H. Hlavacs, M. Haddad, C. Lafouge, D. Kaplan and J. Ribeiro. 2005 The CODIS Content Delivery Network, Computer Networks 48 (1), pp 75-89.

[6] A. Dutta, H. Schulzrinne. 2004 MarconiNet: overlay mobile content distribution network. IEEE Communications Magazine, Volume 42, Issue 2. Pp. 64 - 75.

[7] D. Verma, 2002 Content Distribution Networks. An Engineering Approach, Wiley, New York.

[8] G. Pierre, M. van Steen. 2003 Design and implementation of a user-centered content delivery network, in: Proceedings of the Third IEEE Workshop on Internet Applications, San Jose CA (USA).

[9] Michael J. Freedman, Eric Freudenthal and David Mazières. 2004 Democratizing Content Publication with Coral, in: Proc. 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04), San Francisco, CA.

[10] V.S. Pai, L. Wang, K.S. Park, R. Pang, L. Peterson. 2004 The dark side of the Web: an open proxy's view. ACM SIGCOMM Computer Communication Review.

[11] M. Day, B. Cain, G. Tomlinson, P. Rzewski. 2003 RFC 3466: A Model for Content Internetworking (CDI).

[12] Giancarlo Fortino and Wilma Russo. 2008 Using P2P, GRID and Agent technologies for the development of content distribution networks. Future Generation Computer Systems, Volume 24, Issue 3, Pages 180-190.

[13] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient Overlay Networks," Proc. 18th ACM SOSP, Oct. 2001.

[14] Jian Ni Tsang, D.H.K. Yeung, I.S.H. Xiaojun Hei. Hierarchical content routing in large-scale multimedia content delivery network. IEEE International Conference on Communications, 2003. ICC '03. Vol.2. Pp. 854- 859. May 2003.

[15] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-Aware Overlay Construction and Server Selection," Proc. IEEE INFOCOM. 2002, June 2002.

[16] Wierzbicki, A., Strzelecki, R., Swierczewski, D. and Znojek, M. 2002 Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications, in: Second IEEE International Conference on Peer-to-Peer Computing (P2P2002), Linöping, Sweden.

[17] Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y. 2004 Peer-to-Peer Based Multimedia Distribution Service, IEEE Transactions on Multimedia 6 (2).

[18] Hongjun, L., Luo, L. P. and Zhifeng, Z. 2007 A structured hierarchical P2P model based on a rigorous binary tree code algorithm, Future Generation Computer Systems 23 (2), Pp. 201-208.

[19] Thallner, B., Moser, H. 2005 Topology control for fault-tolerant communication in highly dynamic wireless networks, in: Proceedings of the Third International Workshop on Intelligent Solutions in Embedded Systems.

[20] Yu, J.Y., Chong, P.H.J. "A survey of clustering schemes for mobile ad hoc networks". IEEE Communications Surveys & Tutorials, Vol. 7 (2005), Issue: 1, pp: 32- 48.

[21] J. M. McQuillan, I. Richer, E. C. Rosen. 1980 The New Routing Algorithm for the ARPANET, IEEE Trans. on Communications 28, pp. 711-719.

[22] Jaime Lloret, Juan R. Diaz, Jose M. Jimenez and Fernando Boronat, 2006 An Architecture to Connect Disjoint Multimedia Networks Based on node's Capacity, Lecture Notes in Computer Science 4261, Pp. 890-899.

[23] C. Cramer, K. Kutzner, and T. Fuhrmann. 2004 Bootstrapping Locality-Aware P2P Networks, in: Proc. of IEEE International Conference on Networks, Vol. 1. Pp. 357-361.

[24] Yogen K. Dalal y Robert M. Metcalfe. 1978 Reverse path forwarding of broadcast packets, Communications of the ACM. Volume 21, Issue 12, Pp: 1040-1048.

Improving Mobile and Ad-hoc Networks performance using Group-Based Topologies

Jaime Lloret[1], Miguel Garcia[2] and Jesus Tomas[3]

Department of Communications
Polytechnic University of Valencia
Camino Vera s/n, 46022, Valencia (Spain)
[1]jlloret@dcom.upv.es, [2]migarpi@posgrado.upv.es, [3]jtomas@dcom.upv.es

Abstract. Many works related with mobile and ad-hoc networks routing protocols present new proposals with better or enhanced features, others just compare them or present an application environment, but this work tries to give another point of view. Why don't we see the network as a whole and split it intro groups to give better performance to the network regardless of the used routing protocol?. First, we will demonstrate, through simulations, that grouping nodes in a mobile and ad-hoc networks improves the whole network by diminishing the average network delay and also the routing traffic received by the nodes. Then, we will show which one of the actual fully standardized protocols (DSR [1], AODV [2] and OLSR [3]) gives better performance to the whole network when there are groups of nodes. This paper starts a new research line and urges the researchers to think on it and design group-based protocols.

Keywords: MANET, group-based topologies, network performance.

1 Introduction

The routing protocols in mobile and ad-hoc networks are divided into three types: proactive (which update the routing tables of all the nodes periodically), reactive (which maintain routing routes in their tables only when a node has to communicate with another node in the network) and hybrid (which are a combination of the other two types, taking the advantages of both types). There are many works in the literature that compare the performance of the routing protocols. The most compared protocols have been DSR and AODV. In references [4] and [5] we can see their comparison taking into account some parameters such as the packet delivery fraction, the average delay, the normalized routing load and the throughput consumed when the network load, the mobility and the network size vary. The work in reference [6] added the STAR protocol to the comparison and they measured the data delivery, the control overhead and the data latency. Reference [7] compared DSR and AODV with DSDV taking into account the average delay, the throughput and the control overhead with varied mobility. On the other hand, reference [8] compared DSR, AODV and TORA to analyze the control traffic sent, the data traffic received, the data traffic sent, the throughput, the retransmission attempts, the radio receiver throughput, the radio

---

Please use the following format when citing this chapter:

Lloret, J., Garcia, M. and Tomas, J., 2008, in IFIP International Federation for Information Processing, Volume 264; Wireless Sensor and Actor Networks II; Ali Miri; (Boston: Springer), pp. 209–220.

receiver utilization, the average power, the radio transmitter utilization, the radio transmitter throughput, routing traffic received, routing traffic sent, number of hops and route discovery time. The paper in reference [9] compares the number of packets sent and the traffic sent by DSDV, TORA, DSR and AODV protocols in networks of 50 mobile nodes. Other works compared 5 protocols such as the one presented in [10], where AODV, PAODV, CBRP, DSR and DSDV were compared taking into account the data packet throughput, the average data packet delay and the normalized packet overhead for various number of traffic sources.

Current IETF standardized protocols are AODV [1], DSR [2] and OLSR [3]. None of the works aforementioned have compared them from the group-based topology point of view. We are going to analyze and study their performance when there are group of nodes in their topology.

A cluster is made by a cluster head node, cluster gateways and cluster members. The cluster head node is the parent node of the cluster, which manages and checks the status of the links in the cluster, and routes the information to the right clusters. The rest of the nodes in a cluster are all leaf nodes. The size of the cluster is usually about 1 or 2 hops from the cluster head node. Cluster-based networks are a subset of the group-based networks, because every cluster could be considered as a group. But a group-based network is capable of having any type of topology inside the group, not only clusters. We will take care of group-based topologies in this paper.

The paper is structured as follows. Section 2 shows and analyzes the differences between DSR, AODV and OLSR protocols when regular and group-based topologies are used. The group-based topologies comparison is shown in section 3. Finally, section 4 gives our conclusions.

2 Group-based topology performance

2.1 Test Bench

This sub-section presents the test-bench used for all the evaluated protocols. The number of nodes and the coverage area of the network have been varied. Each protocol has been simulated in 4 scenarios: (1) With fixed nodes, (2) With mobile nodes and failures, (3) With grouped nodes and (4) With grouped mobile nodes and failures. Each scenario has been simulated for 100 and 250 nodes, to observe the system scalability. Instead of a standard structure we have chosen a random topology. Figure 1 shows the 100 nodes topology (in a 750x750 m^2 area) and Figure 2 shows the 250 nodes topology (in a 1 Km^2 area). It has been obtained using the version Modeler of OPNET simulator [11]. Both topologies have been created using different seeds. Arrows indicate that nodes are mobile and change their position constantly. The green lines from each node (blue circles) indicate the node mobility. We can see that the nodes are inside a blue box. This box shows a wireless area and it has been used to delimit the mobility area of the nodes. In that area, a node can move randomly during the simulation. The physical topology doesn't follow any known pattern. The obtained data don't depend on the initial topology of the nodes or on their movement pattern, because all of it has been fortuitous.

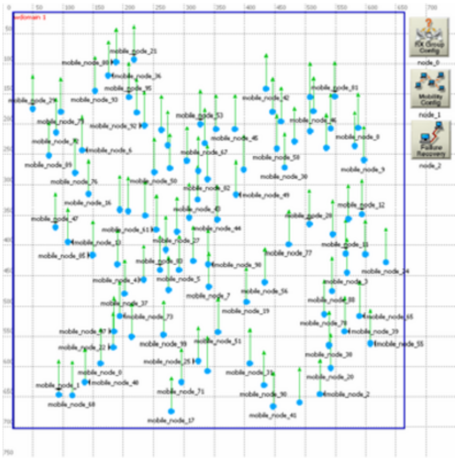


Fig 1. Topology with 100 nodes.

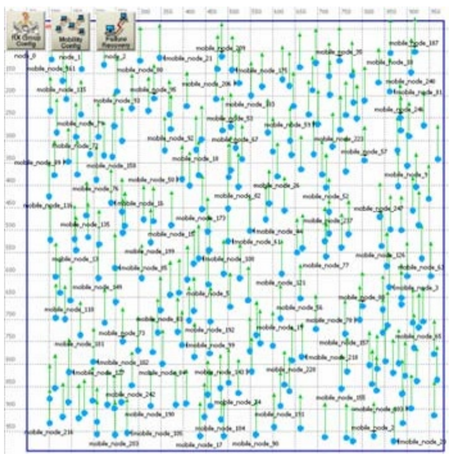


Fig 2. Topology with 250 nodes.

We have created 6 groups for the 100 nodes topology, covering approximately, a circular area with a 150 meter radius each group. There are 16 or 17 nodes approximately, in each group. The number of nodes in each group varies because of the node's random mobility. A node can change a group anytime. For the 250 nodes topology, we have created 12 groups, with 15 or 16 nodes per group approximately covering a circular area with a 150 meter radius each group.

The ad-hoc nodes of the topologies have a 40 MHz processor, a 512 KB memory card, a radio channel of 1 Mbps and their working frequency is 2.4 GHz. Their maximum coverage radius is 50 meters. This is a conservative value because most of the nodes in ad-hoc network have larger coverage radius, but we preferred to have lower transmitting power for the ad-hoc devices to enlarge their time of life.

We have forced node failures at t=200 sec., t=400 sec. and t=1200 sec. in each network, with a recovering process of 300 sec., to take measurements from the mobile nodes simulation when the physical topology changes.

The MANET traffic generated by OPNET has been used as the simulations' traffic load. We inject this traffic 100 seconds after the simulation starts. We have configured the traffic arrival with a Poisson distribution (with a mean time between arrivals of 30 seconds). The packet size follows an exponential distribution with a mean value of 1024 bits. The destination address of the injected traffic is random to obtain a simulation independent of the traffic direction. We have simulated the four scenarios for DSR, AODV and OLSR protocols. The results obtained are shown in the following sub-sections.

2.2 DSR, AODV and OLSR in group-based topologies

Figures 3 and 4 show the average delay of the DSR protocol in fixed and mobile topologies at the application layer. In figure 3 we observe that group-based topologies have an average delay close to 0.005 seconds regardless of the number of nodes in the network. In the regular network the delay has a value of 0.02 seconds for 100-nodes topology and of 0.03 seconds for the 250-nodes topology when the network converges. In the case of the 100-nodes topology there is an improvement of 75% and it is better in the 250-nodes topology (an 83% of improvement). The topologies with mobility and errors (figure 4) shows that the average delays at the application layer are higher in the group-based topologies till the network converges. Although group-based topologies present worse behaviour till 1300 seconds, when the network is stabilized, group-based topologies have an improvement around 5%.

Then, we have compared the routing traffic received in the DSR protocol (figures 5 and 6). Figure 5 shows that the traffic is quite stable due to the characteristics of the network. It is due to it is a fixed network without errors and failures. The traffic received in the 250-node topology is around 500 Kbits/s, but when we group the nodes this traffic decreases until 200 Kbits/s (a 60% of improvement). The value obtained in a 100-node topology (250 Kbits/s), is also improved when we group the nodes (100 Kbits/s), therefore there is a 60% of improvement. In figure 6 we observe a similar behaviour. In this case we conclude that when there are errors and failures in the 250-nodes topology the traffic fluctuates and is less stable (we can observe it in the intervals from 600 to 800 seconds and around 1200 seconds). We also observe that the instability is much lower in group-based topologies. 100-nodes topology has a mean value around 175 Kbits/s, while 100-nodes group-based topology has a mean value around 95 Kbits/s, so there is an improvement of 46%. On the other hand, 250-nodes topology has a mean value around 400 Kbits/s, while 250-nodes group-based topology has a mean value around 180 Kbits/s, so there is an improvement of 55%.

The average delay at the application layer in the AODV protocol can be seen in figures 7 and 8. Both topologies, 100-nodes and 250-nodes, give an average delay higher than 0.5 seconds when the network converges, but there are some peaks higher than 2.5 seconds. On the other hand, group-based topologies have a similar delay which is around 0.15 seconds. Group-based topologies improve the delay at the application layer in 70%. When the topology with mobile nodes is used, the simulation shown in figure 8 is obtained. In case of 250 nodes, there is a delay of 1 second when the network has converged. The case of 100 nodes gives an average delay of 0.75 seconds approximately. When there are group-based topologies, the delay decreases to 0.25 seconds in both cases. There is an improvement of 75% for the 250-nodes topology and 67% for the 100-nodes topology.

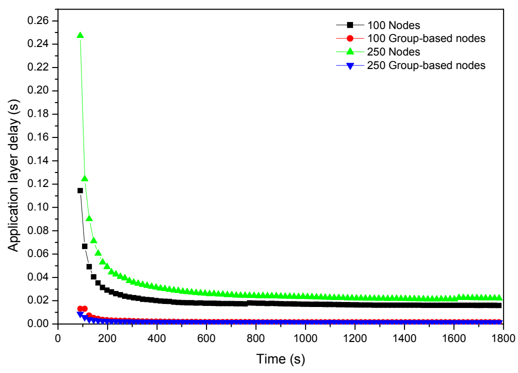


Fig. 3. DSR average delay at the application layer in fixed topologies.

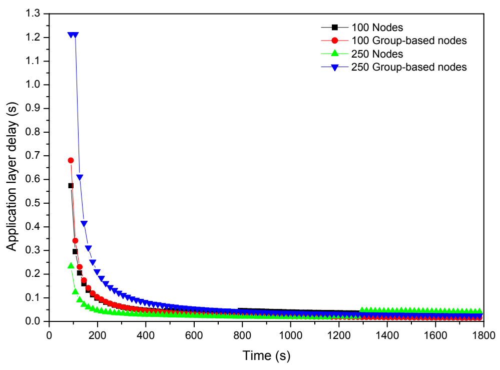


Fig. 4. DSR average delay at the application layer in mobile topologies.

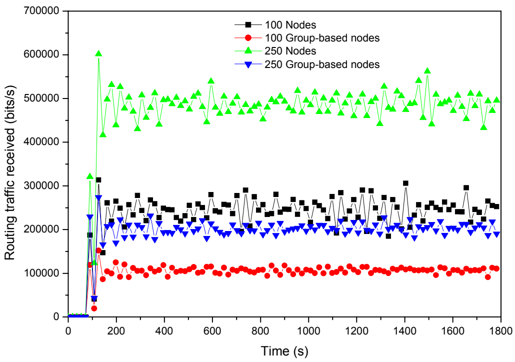


Fig. 5. DSR routing traffic received in fixed topologies.

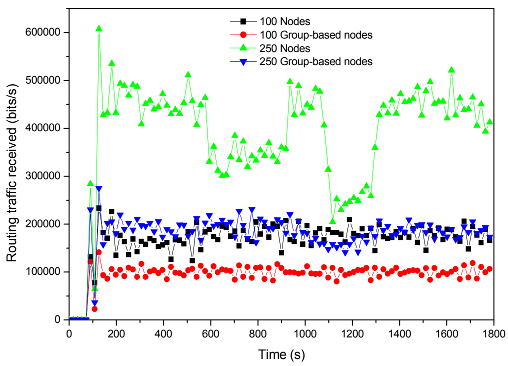


Fig. 6. DSR routing traffic received in mobile topologies.

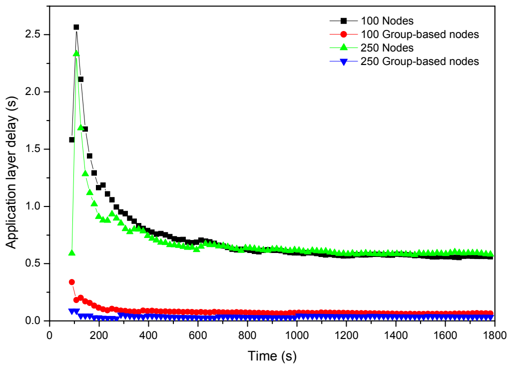


Fig. 7. AODV average delay at the application layer in fixed topologies.

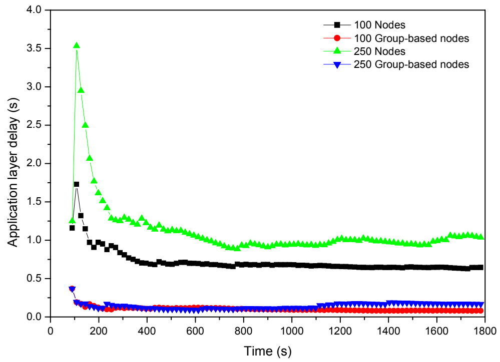


Fig. 8. AODV average delay at the application layer in mobile topologies.

The routing traffic received for the AODV in each simulated topology can be seen in figures 9 and 10. We observe that the routing traffic received is independent of the mobility of the nodes. In figure 9 we can see that the routing traffic goes from 440 Kbits/s for 250-node case to 250 Kbits/s when there are group of nodes (a 43% of improvement). In the 100-node case, it goes from 230 Kbits/s to 140 Kbits/s when it is a group-based topology (a 39% of improvement). When there are mobility and errors and failures (see figure 10), in the 250-node topology the values go from 440 Kbits/s to 250 Kbits/s in the group-based topology (a 43% of improvement). We obtained 200 Kbits/s in the regular 100-node topology and 135 Kbits/s for the group-based one (a 32% of improvement).

In figure 11, the delay at the application layer simulated for the OLSR protocol using fixed topologies is shown. In the case of 250 nodes we have obtained a delay around 0.015 seconds, which has changed to 0.0035 seconds in the case of 250-nodes group-based topology (there is a 76% of improvement). In the case of 100 nodes, it has decreased from 0.005 seconds in the regular topology to 0.002 seconds in the group-based topology, so there is a 60% of improvement. When there is mobility and errors and failures in the network for the OLSR protocol (see figure 12), we observe that the 100-nodes regular topology has a delay at the application layer of 0.007 seconds when the network has converged, but there is a delay of 0.0025 seconds for the 100-nodes group-based topology (a 64% of improvement). In the case of 250 nodes the improvement is around 60 %. We have obtained a delay of 0.005 seconds in the regular topology versus 0.002 seconds in the group-based topology.

Finally, we have studied the behaviour of the OLSR protocol analyzing the mean routing traffic received (figures 13 and 14). The routing traffic received in the 100-node fixed topology was around 180 Kbits/s, while in group-based topology has decreased to 70 Kbits/s, so there is a 61% of improvement. In the 250-node topology case, we appreciate that this traffic was approximately 300 Kbits/s, but there are values lower than 150 Kbits/s in the group-based topology (figure 13). So there is a 50% of improvement. Figure 14 shows the results of a network with mobility and errors and failures. We have observed some fluctuations due to the failures and errors in the network, in both 100-node and 250-node topologies. Those fluctuations are minimized when we use group-based topologies. Improvements of 61% and 50% are obtained in 100-node and 250-node topologies, respectively.

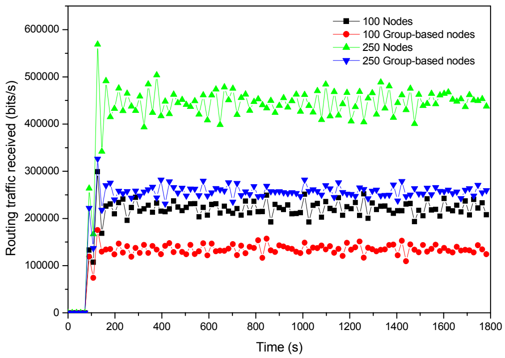


Fig. 9. AODV routing traffic received in fixed topologies.

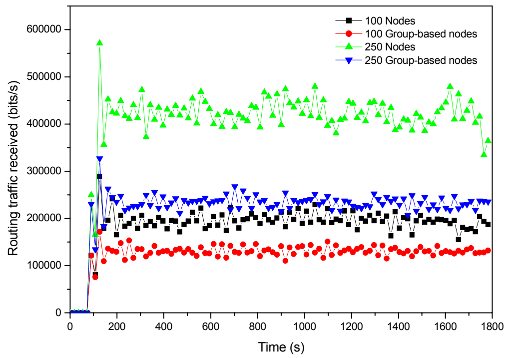


Fig. 10. AODV routing traffic received in mobile topologies.

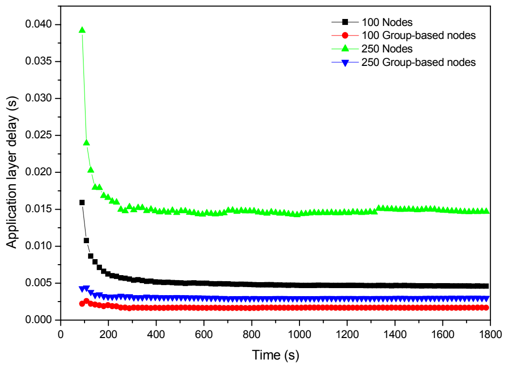


Fig. 11. OLSR average delay at the application layer in fixed topologies.

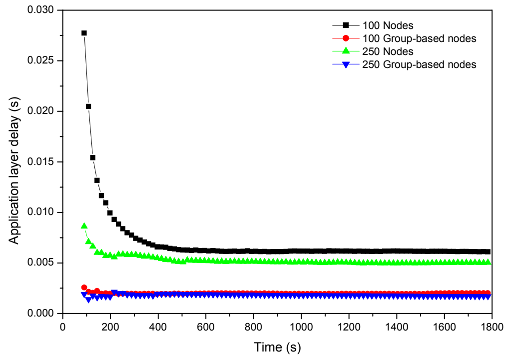


Fig. 12. OLSR average delay at the application layer in mobile topologies.

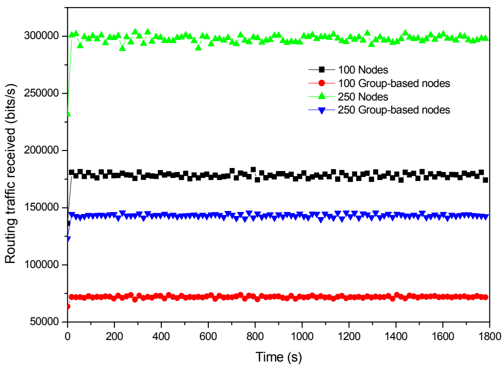


Fig. 13. OLSR routing traffic received in fixed topologies.

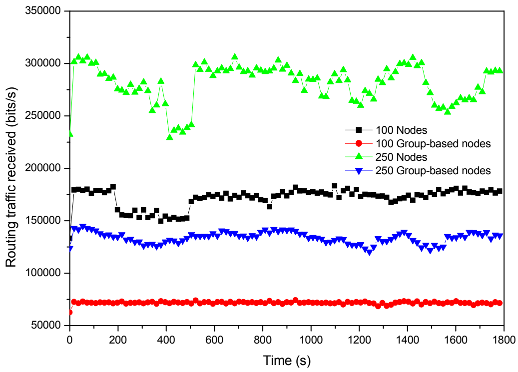


Fig. 14. OLSR routing traffic received in mobile topologies.

3 Group-based topologies comparison

In order to make the comparison of DSR, AODV and OLSR using group-based topologies, we have used the same test bench used in section 2. This comparison will show us which mobile and ad-hoc routing protocol have better features for group-based topologies.

The average delay at MAC layer in fixed group-based topologies is shown in figure 15. All routing protocols have an average delay lower than 0.001 seconds when the network has converged in both 100-nodes and 250-nodes topologies. It shows that group-based topologies have a good behaviour. DSR protocol with 100-node topology has been the one with worst behaviour and OLSR in 250-node topology has been the best one. OLSR protocol has the same delay (around 0.001 seconds) for both topologies, 100-nodes and 250-nodes, approximately, and it is the most stable. Figure 16 shows the simulation for mobile and errors and failures topologies. All protocols have a delay lower than 0.001 seconds when the network has converged. In this case, AODV protocol has the worst behaviour and OLSR protocol is the most stable.

When the average throughput consumed in the fixed group-based topologies is compared (Figure 17), the protocol that consumes lowest throughput is the DSR protocol (90 Kbits/s in the 100-node topology and 170 Kbits/s in the 250-node topology). The protocol with the most stable throughput consumed is the OLSR protocol. When the network converges, both AODV and OLSR protocols have the same average throughput in the 100-nodes topology, but the OLSR protocol has the lowest convergence time. In case of having a group-based topology with mobility and errors and failures (see figure 18), the results are very similar to the previous ones. The protocol that consumes lower throughput is DSR. AODV protocol consumes lower throughput while the network is converging, but this throughput becomes very similar to the one given by OLSR protocol when the network converges. OLSR protocol is still the most stable.

Then, we analyzed the protocols behaviour when there is MANET traffic. In fixed group-based topologies (see figure 19), the 250-nodes topology shows that the protocol with lower traffic is AODV (40 bits/s approximately) and the one with highest traffic is OLSR. In the 100-nodes topology all protocols have similar behaviour (between 160 bit/s and 180 bits/s). When the network has converged, we can consider AODV and DSR as the best ones and OLSR as the worst. When there is mobility in the group-based topology (see figure 20), the protocol with lowest MANET traffic in the 250-nodes topology is DSR protocol (80 bits/s approximately) and the worst is OLSR protocol. In the case of 100-node topology the one with lowest MANET traffic is DSR protocol and the worst OLSR.

When we analyze the routing traffic sent in fixed group-based topologies (see figure 21) we observe that the one which sends more routing traffic is AODV protocol, (around 120 Kbit/s in the 250-nodes group-based topology and 56 Kbits/s in the 100-nodes group-based topology). OLSR protocol has the best behaviour. It is more stable than the other ones and it sends lower routing traffic than the others (64 Kbits/s in case of the 250-nodes topology and 28 Kbits/s in the 100-nodes topology). When we analyze the mobile group-based topology (figure 22), although the routing traffic has decreased very few, the behaviour of the protocols is very similar to the fixed group-based topologies (Figure 21). AODV is the worst protocol because it is the one which sends more routing traffic to the network and OLSR is the most stable and the one which sends lower routing traffic to the network. The one which has worst stability in mobile group-based topologies is the DSR protocol.

The routing traffic sent is obtained by measuring every node as a source and figures 21 and 22 give the whole routing traffic sent by all of them. However, the routing traffic received is obtained by adding the traffic received by all nodes. The routing traffic received in fixed and mobile group-based topologies is shown in figures 23 and 24 respectively. We can see that it is more than the double of the values obtained for the routing traffic sent. In fixed group-based topologies (see Figure 23) AODV protocol is the one that gives higher routing traffic received (around 250 Kbit/s in 250-nodes topology and 135 Kbits/s in 100-nodes topology). OLSR protocol is the most stable and the one with lower routing traffic received (145 Kbits/s in 250-nodes topology and 70 Kbits/s in 100-nodes topology). When the mobile group-based topologies are analyzed (figure 24), AODV protocol is the one that has worst behaviour and OLSR is the most stable and the one that has lower routing traffic sent. DSR protocol is the most instable.

Figure 25 shows the average delay at application layer in fixed group-based topologies. The protocol most instable and with higher delay in 100-nodes and 250-nodes topologies is AODV protocol. It has peaks with more than 0.45 seconds and it is stabilized around 1700 seconds with a mean value of 0.15 seconds. DSR and OLSR are the ones with lowest delay. Figure 26 shows the average delay at application layer in mobile group-based topologies. DSR protocol is the one that has worst delay till the network converges. Then, when the network is stabilized, the worst is AODV protocol which has delays between 0.1 and 0.15 seconds. OLSR protocol gives the lowest delays.

Then, we have compared DSR and AODV in some common reactive protocols features. In figure 27 the average number of hops in a path for fixed group-based topologies can be observed. DSR protocol has an average value of hops close to 5 in the 250-nodes topology when the network has converged. The number of hops in the 100-nodes topology is slightly lower. AODV has lower average number of hops (around 3.25 hops in the 250-nodes case and 2.75 in the 100-nodes case). The convergence time for the DSR protocol is quite lower than AODV, but it is more instable. In the case of mobile group-based topologies (see figure 28) the behaviour is similar as the previous one, so there is not any dependence on the mobility.

Now we have analyzed the route request sent in reactive protocols for fixed and group-based topologies (figures 29 and 20 respectively). AODV protocol is the one with most number of route requests sent (860 approximately in 250-nodes topology and 330 approximately in 100-nodes topology). We have observed a relationship between the number of route requests sent in the AODV protocol and number of nodes in the topology. There is approximately a factor of 3.3. In the DSR protocol, the number of route requests sent is equal to 730 in the 250-nodes topology and 190 in the 100-nodes topology. Both, fixed and mobile, present the same behaviour. We have observed that the route request sent is the only parameter that gives worst values in group-based topologies than in regular topologies.

Table 1 shows the best and worst protocols for all parameters analyzed.

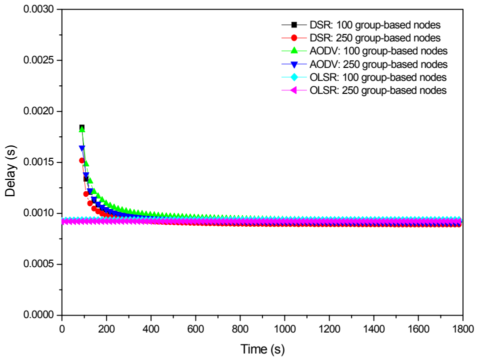


Fig. 15. Comparison of average delays at MAC layer in fixed topologies.

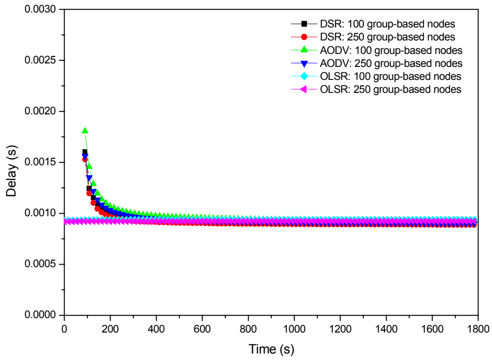


Fig. 16. Comparison of average delays at MAC layer in mobile topologies.

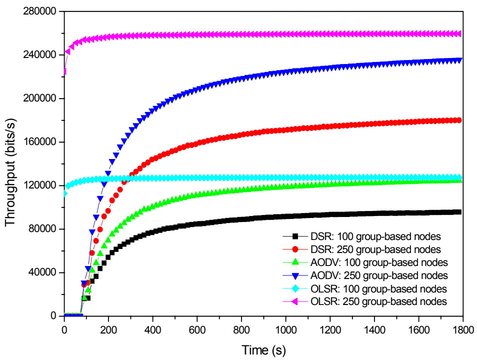


Fig. 17. Comparison of average throughputs consumed in fixed topologies.

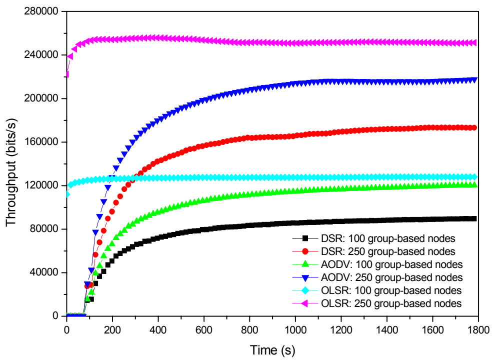


Fig. 18. Comparison of average throughputs consumed in mobile topologies.

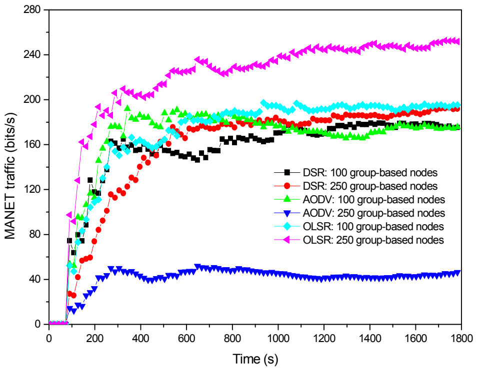


Fig. 19. Comparison of average MANET traffic in fixed topologies.

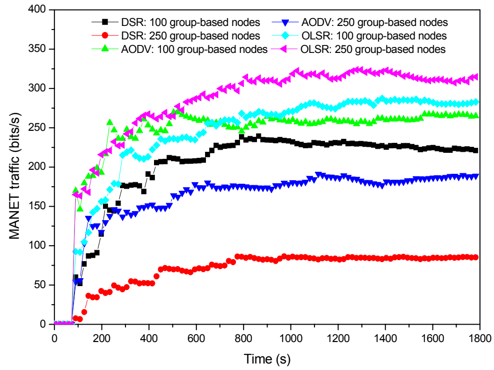


Fig. 20. Comparison of average MANET traffic in mobile topologies.

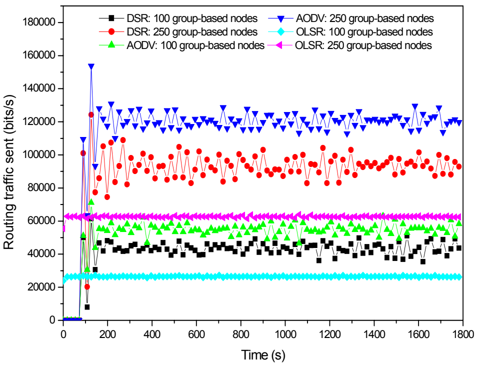


Fig. 21. Comparison of routing traffic sent in fixed topologies.

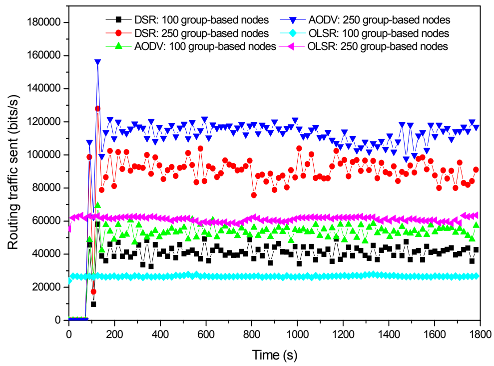


Fig. 22. Comparison of routing traffic sent in mobile topologies.

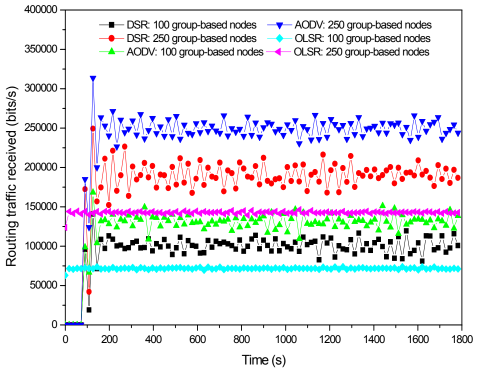


Fig. 23. Comparison of routing traffic received in fixed topologies.

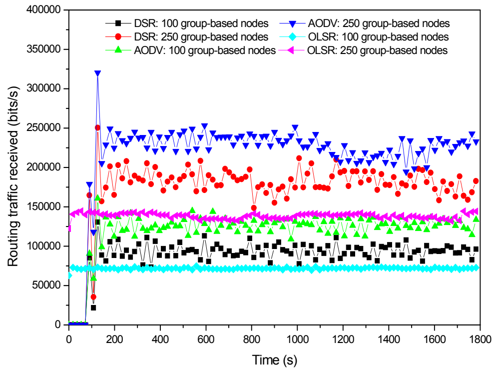


Fig. 24. Comparison of routing traffic received in mobile topologies.

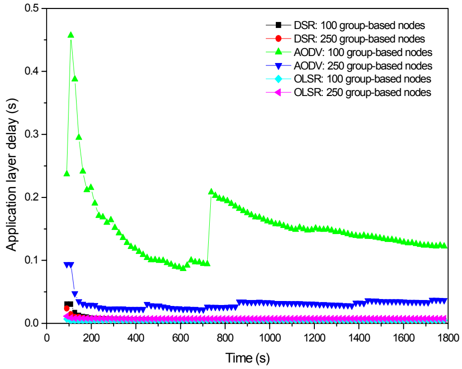


Fig. 25. Comparison of the average delay at application layer in fixed topologies.

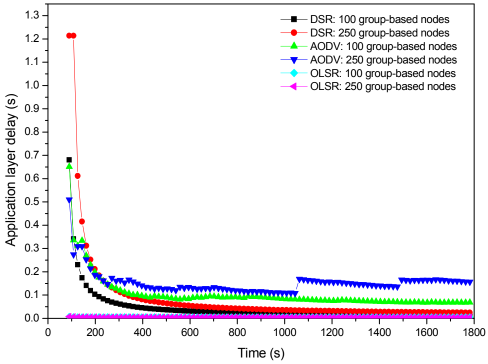


Fig. 26. Comparison of the average delay at application layer in mobile topologies.

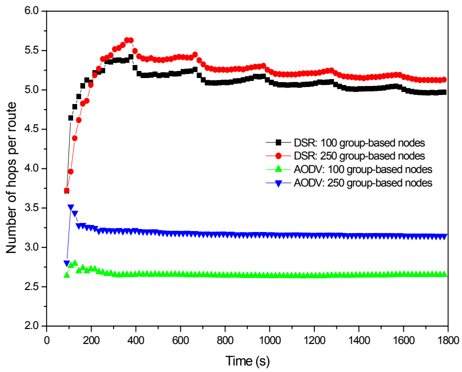


Fig. 27. Comparison of the average number of hops in a path in fixed topologies.

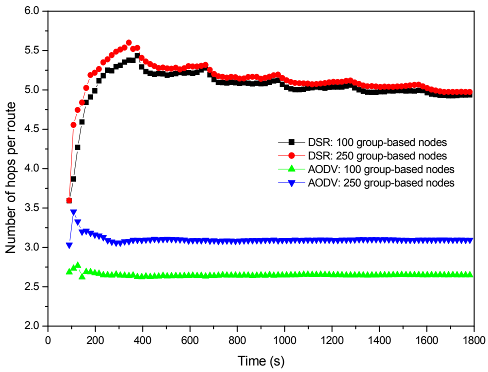


Fig. 28. Comparison of the average number of hops in a path in mobile topologies.

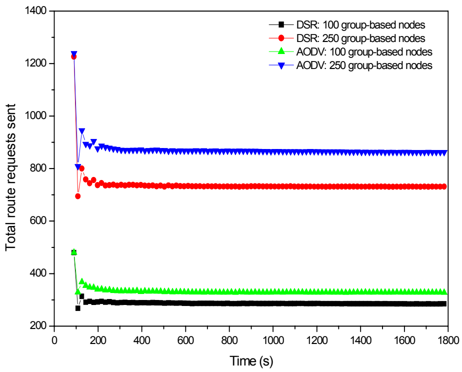


Fig. 29. Comparison of the average number of route requests sent in fixed topologies.

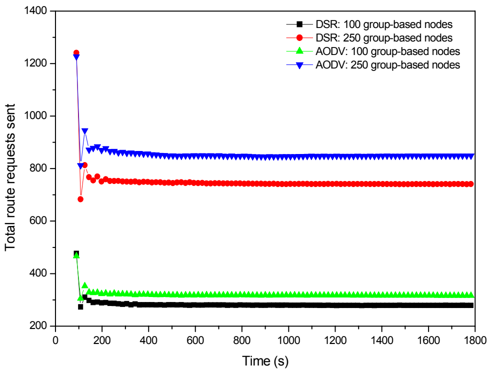


Fig. 30. Comparison of the average number of route requests sent in mobile topologies.

Table 2. Comparison of mobile and ad-hoc routing protocols in group-based topologies.

	Best in fixed	Best in mobile	Worst in fixed	Worst in mobile
Delay at MAC layer	OLSR	OLSR	DSR	AODV
Throughput consumed	DSR	DSR	AODV & OLSR	AODV & OLSR
MANET traffic	AODV	DSR	OLSR	OLSR
Routing traffic sent	OLSR	OLSR	AODV	AODV
Routing traffic received	OLSR	OLSR	AODV	AODV
Delay at application layer	DSR & OLSR	OLSR	AODV	AODV
Average number of hops in a path	AODV	DSR	AODV	DSR
Route requests sent	DSR	AODV	DSR	AODV

4 Conclusions

We have simulated 3 MANET routing protocols with grouping nodes, to demonstrate that group-based topologies improve the network performance. The best improvement percentage has been the DSR protocol when the average delay at the application layer has been simulated. We have observed more improvement in fixed topologies when there are 250 nodes in the topology, but when there is a mobile topology, the improvement is higher in the topology with 100 nodes. When a routing protocol is the best one in a fixed group-based topology, it continues being the best one in the mobile group-based topology. On the other hand, we have observed that a routing protocol, which is the best (or worst) in a group-based fixed topology, could not be the best (or worst) in the mobile topology. The routing protocol that has appeared more as the best one has been OLSR and the one that has appeared as the worst one has been AODV.

References

1. D. Johnson, Y. Hu, D. Maltz. "The Dynamic Source Routing Protocol (DSR) for Mobile Ad-hoc Networks for IPv4". RFC 4728. February, 2007.
2. C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing". RFC 3561. July, 2003.
3. T. Clausen and P. Jacquet. "Optimized Link State Routing Protocol". RFC 3626. Oct. 2003.
4. R. Misra, C. R. Mandal, “Performance comparison of AODV/DSR on-demand routing protocols for ad hoc networks in constrained situation”, IEEE International Conference on Personal Wireless Communications, 2005. Pp. 86- 89. January 2005
5. G. Jayakumar, G. Ganapathy “Performance Comparison of Mobile ad-hoc Network Routing Protocol”. Int. Journal of Computer Science and Network Security, Vol.7 No.11, Nov. 2007.
6. H. Jiang and J. J. Garcia-Luna-Aceves, “Performance comparison of three routing protocols for ad hoc networks”. Proceedings of IEEE ICCCN 2001.
7. S. Ahmed, M. S. Alam, “Performance Evaluation of Important Ad Hoc Network Protocols”. EURASIP Journal on Wireless Communications and Networking. Vol. 2006. Pages 1–11.
8. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks”. Proc. of the 5th ACM/IEEE int. conf. on Mobile computing and networking. Pp. 195-206. 1999.
9. J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols". Proc. of the 4th ACM/IEEE int. conf. on Mobile computing and networking. Pp. 85-97. 1998.
10. Azzedine Boukerche, “Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks”, Journal of Mobile Networks and Applications. Vol. 9, No 4. Pp. 333-342, 2004.
11. OPNET Modeler website. At, http://www.opnet.com/solutions/network_rd/modeler.html

Using MANET protocols in Wireless Sensor and Actor Networks

Miguel Garcia[1], Hugo Coll[2], Diana Bri[3], Jaime Lloret[4]
Department of Communications, Polytechnic University of Valencia, SPAIN
[1,2,3]{migarpi, hucolfer, diabrmo}@posgrado.upv.es, [4]jlloret@dcom.upv.es

Abstract

Although there are several routing protocols for Wireless Sensor and Actor Networks (WSAN), none of them have became standard. Now, there are several standard protocols for Mobile Ad hoc Networks (MANET) that have been developed for devices with higher computing features than the sensor nodes. On the other hand, one of the main characteristics of the MANET protocols is their scalability. In this paper, we show the performance of a WSAN, when MANET protocols are used, for several topologies. We will discuss and evaluate which standard protocol is the best one depending on the number of nodes in the topology and depending on their mobility. Finally we will show their comparison. As far as we know, there is not any performance comparison of MANET protocols such as the one presented in this paper.

1. Introduction

Wireless technologies are becoming important in the world of communications. It is mainly due to three reasons: mobility, low cost and good bandwidth.

MANET networks [1] do not have infrastructure or any access point. This is a feature that makes difficult the communication between the participants of the network. Moreover, the allocation of IP addresses to build routes where the information will travel from a source node to a destination node is so difficult.

The WSAN [2] are distributed networks where intermediate nodes are used to route the information. Sensor networks are taking relative importance in many daily activities since some years ago [3]. Some of the applications where they are using this type of technologies are [4]: a) Industrial control and monitoring, b) Home automation and consumer electronics, c) Security and military sensing, d) Asset tracking and supply chain management, e) Intelligent agriculture and environmental sensing and f) Health monitoring.

The main difference between MANET and WSAN is that MANET devices are designed to be in contact with the human, while the WSAN devices do not interact directly with the human, so the devices used in both networks are different. Although, the nodes of both types of networks can be placed in hostile environments where the network topology changes constantly and where there is a high probability of node failure. The protocols developed in WSANs must be designed for failure tolerance and node energy savings. In short, the protocols used in WSANs must take into account to save energy, so they are based on the following features:

- The protocols and the node's operations are based on cycles.
- The radio coverage for each sensor node usually has between 50 and 75 metres, in the free space.
- The processing capacity and the transmission rate of the nodes are low.

In this paper we will demonstrate that doing an adequate sizing of the network we can use MANET protocols in wireless sensor networks.

The structure of the paper is as follows. Related works are presented in the next section. Section 3 describes the MANET protocols that are going to be used in this paper. Our simulations, analysis and comparison are shown in section 4. Finally, in section 5, we conclude the paper and give our highlights.

2. Related Works

There are many works in the literature that compare the performance of the routing protocols. The most compared protocols have been DSR and AODV. In references [5] and [6] we can see their comparison taking into account some parameters such as the packet delivery fraction, the average delay, the normalized routing load and the throughput consumed when the network load, the mobility and the network size vary. The work in reference [7] added the STAR protocol to the comparison and the authors measured the data delivery, the control overhead and the data latency.

978-0-7695-3330-8/08 $25.00 © 2008 IEEE
DOI 10.1109/SENSORCOMM.2008.102

Reference [8] compared DSR and AODV with DSDV taking into account the average delay, the throughput and the control overhead with varied mobility. On the other hand, reference [9] compared DSR, AODV and TORA to analyze the control traffic sent, the data traffic received, the data traffic sent, the throughput, the retransmission attempts, the radio receiver throughput, the radio receiver utilization, the average power, the radio transmitter utilization, the radio transmitter throughput, routing traffic received, routing traffic sent, number of hops and route discovery time. Other works compared 4 protocols, such as the one presented in [10] where they compared the number of packets sent and the traffic sent by DSDV, TORA, DSR and AODV protocols in networks of 50 mobile nodes. And others even 5 protocols such as the one presented in [11], where AODV, PAODV, CBRP, DSR and DSDV were compared taking into account the data packet throughput, the average data packet delay and the normalized packet overhead for various number of traffic sources.

There are other works where the routing protocols in WSANs are discussed. E.g., in [12] there is a detailed study of many routing protocols. This study explains the operation mode of all protocols exposed and, at the end of the paper, presents a classification and a comparison. The comparison if given by means of their features, but it is not a study on the amount of control traffic introduced, delay, throughput rate, etc.

In many studies about wireless sensor networks [13], authors evaluate many routing protocols by the increase in the length of life, energy saving, etc.

We have not found any work where the performance of the routing protocols in WSANs is studied. Because of it, in this work we evaluate the performance of MANET protocols when they are used in different WSAN scenarios.

3. MANET Protocols

IETF (Internet Engineering Task Force) MANET working group (see [1]) is responsible to analyze the problems in the ad hoc networks and to observe their performance. This work group classifies the MANET protocols in three large groups. The first one is the proactive protocols, which periodically update the routing tables of all nodes of the network, although they could not be sending information (e.g. OLSR [14] and TBRPF [15]). The second group is formed by the reactive protocols. They are based on the calculation of the optimum route between a source node and a destination node when a node wants to perform a communication (e.g. AODV [16] and DSR [17]). The third group is formed by the hybrids protocols, which combine some features of the other two protocols (ZRP [18]).

In the following subsections we are going to explain briefly the three most used MANET standard protocols. These three protocols (AODV, DSR and OLSR) together with TBRPF have their own RFC.

3.1. AODV

AODV (Ad hoc On-demand Distance Vector) [16] is a proactive protocol. In this protocol, the nodes use the sequence numbers to avoid loops and take the path information as updated as possible. When a source node wants to transmit information to a destination node, it sends a RREQ (Route Request) packet in broadcast mode to request a route. If a node sees that it is in the destination field of a RREQ, first it checks that this packet has not been received yet by means of a RREQ register. If it was not registered, it sends the message back and increases the number of hops and creates the route reverse replying with a RREP (Route Reply) packet to confirm the path. For the maintenance of the routes can be used 2 methods: a) ACK messages in MAC level or b) HELLO messages in network layer.

3.2. DSR

The reactive protocol DSR (Dynamic Source Routing Protocol) [17] is a protocol specifically created for ad hoc networks. The route discovery is done when the source node does not have in its routing table the path to the destination node. In order to discover it, it sends a RREQ packet in broadcast mode. When a node receives a RREQ packet with a destination that is in its routing table, it sends the RREP packet with a copy of the accumulated route registration of the RREQ packet plus the route to that node in its routing table. When the intermediate node does not know the destination, it must enter its address in the accumulated route registration of the RREQ packet and it forwards the RREQ message in broadcast mode. When a node learns the route, this is stored in a temporal cache. The maintenance of the route is conducted through passive acknowledgements in data link level and acknowledgements in network layer.

3.3. OLSR

OLSR (Optimized Link State Routing Protocol) [14] is proactive protocol. This feature provides a fast routing protocol, in addition to the advantage of knowing all nodes and IP addresses in the network. But nodes require more memory. Its operation mode is as follows. Each node selects its set of MPR (Multi Point Relay) nodes, which will be responsible for sending the traffic to the network and to avoid the duplicate packets. Neighbour nodes and the MPR nodes assignation are learned using hello messages. The Topology Control messages send information about the network topology. The OLSR operation is very similar to the OSPF protocol in wired networks.

4. Simulations

We are going to present the simulations done to measure the amount of traffic in the WSAN when MANET routing protocols are used.

4.1. Test bench

This sub-section presents the test-bench used for all the evaluated protocols. We have varied the number of nodes and the coverage area of the network. Each protocol has been simulated in 2 scenarios: 1) With fixed nodes and 2) With mobile nodes and failures.

Each scenario has been simulated for 100 nodes (in a 750x750 m^2 area) and 250 nodes (in a 1 Km^2 area), to observe the system scalability. Instead of a standard structure we have chosen a random topology. It has been obtained using the version Modeler of OPNET simulator [19]. The nodes have a random mobility model. The physical topology does not follow any known pattern. The obtained data do not depend on the initial topology of the nodes or on their movement pattern, because all of it has been fortuitous.

The nodes have a 40 MHz processor, a 512 KB memory card, a radio channel with less than 1 Mbps and their working frequency is 2.4 GHz. Their maximum coverage radius is 50 meters.

We have forced node failures, with the consequent recovering processes, to take measurements from the mobile nodes simulation and to observe the network behaviour against physical topology changes.

The traffic generated by OPNET has been used as the simulations' traffic load. We injected this traffic 100 seconds after the simulation starts. We have configured the traffic arrival with a Poisson distribution (with a mean time between arrivals of 30 seconds). The packet size follows an exponential distribution with a mean value of 1024 bits. The destination address of the injected traffic is random to obtain a simulation independent of the traffic address. We have simulated both scenarios for AODV, DSR and OLSR protocols. The results obtained are shown in the following sub-sections.

4.2. Routing traffic

This subsection shows the routing traffic obtained from both scenarios.

In Fig. 1 the routing traffic for 100 and 250 fixed nodes topologies is shown. We can observe that the OLSR protocol is the one which introduces least routing traffic into the network in both topologies. We have a mean routing traffic routing equal to 58.5 Kbits/s in the topology of 100 nodes and 112.2 Kbits/s in the case of 250 nodes. AODV and DSR protocols introduce more routing traffic. DSR protocol had worse behaviour than the AODV. In the 100 nodes topology, the mean routing traffic introduced by AODV is equal 84.8 Kbits/s while the DSR introduces an average rate of 94.1 Kbits/s. In the case of 250 nodes this traffic is increased. The AODV brings 182.1 Kbits/s and the DSR has 201.7 Kbits/s. OLSR is the most stable protocol. As the number of nodes is increased the relationship routing traffic vs. number of nodes is decreased.

Fig. 2 shows the simulation measurements of the 100 and 250 nodes topologies but when there are random mobility and node failures and recoveries. In this case, the routing traffic decreases because there are fewer nodes sending information. But when there are recoveries, the traffic increases to seek new routes. The OLSR protocol is the one which provides lowest traffic load (56.9 Kbits/s for 100 nodes and 105.6 Kbits/s with 250 nodes) and with greatest stability. DSR protocol had the worse behaviour. This protocol introduces more network traffic and when there are errors in the network it has several very abrupt fluctuations. When the number of nodes increases the behaviour of the network is worst.

4.3. Throughput

The study of the average throughput is shown in Fig. 3 and Fig. 4.

In Fig. 3, the average throughput consumed into the network for each protocol (100 and 250 nodes topologies in the fixed nodes scenario) is shown. The OLSR protocol is the one which consumes a higher throughput rate. It happens because OLSR is a proactive protocol and it sends control messages periodically. OLSR has the average throughput of 332.2 Kbits/s in the 100 nodes topology and 544.2 Kbits/s for 250 nodes. Although this protocol consumes highest throughput rate, it is the most stable. In this case the protocol which has the best behaviour is AODV. It has a throughput consumption rate of 210 Kbits/s in the 100 nodes topology when the network converges and 415 Kbits/s in the case of 250 nodes.

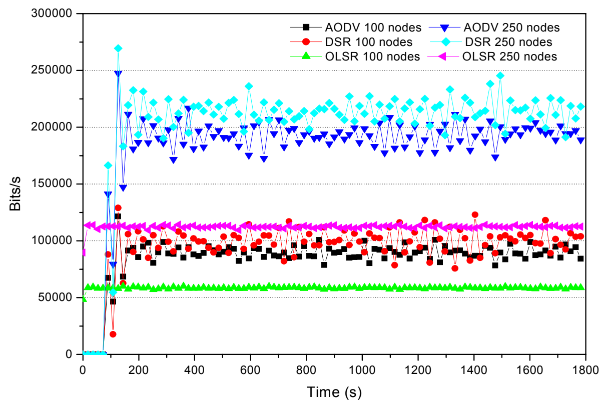


Fig. 1. Routing traffic in fixed nodes.

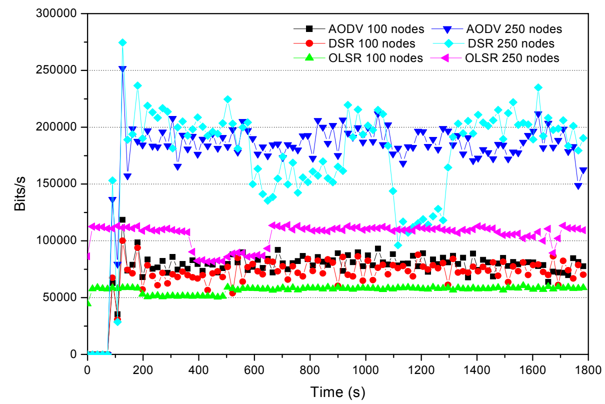


Fig. 2. Routing traffic in mobile nodes and failures

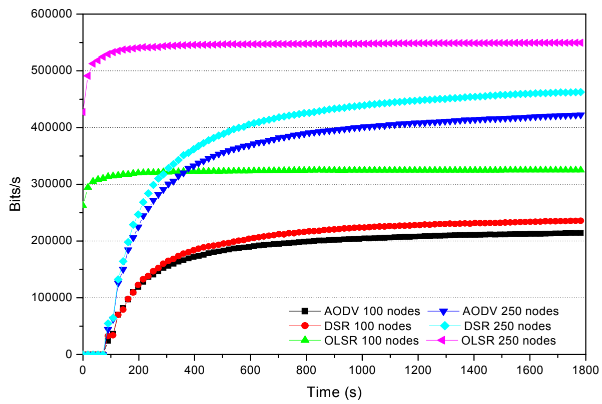


Fig. 3. Average throughput in fixed nodes

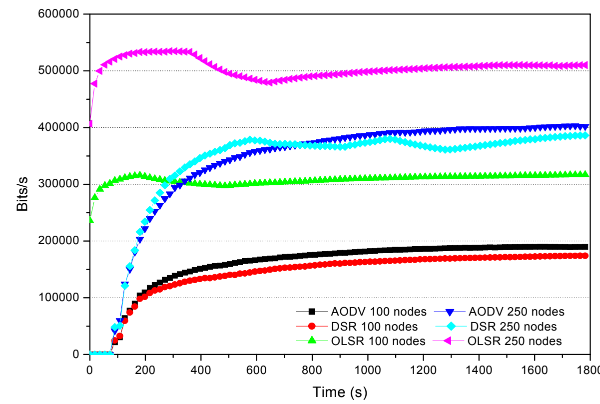


Fig. 4. Average throughput in mobile nodes.

Fig. 4 shows the average throughput in the topologies with 100 and 250 nodes for mobile nodes with errors. In this case, OLSR is the protocol with more throughput consumption. It is the most stable protocol. We obtained a throughput rate of 308 Kbits/s in the 100 nodes topology and 505 Kbits/s for the 250 nodes. So, when the nodes have mobility, DSR protocol is the best one (170 Kbits/s for 100 nodes and 380 Kbits/s in the 250 nodes).

4.4. Delay

In this subsection we observe the behaviour that has the protocols from the point of view of the delay.

Fig. 5 shows the average delay in the 100 and 250 nodes topologies for the fixed scenarios. The analysed protocols have a very low delay (below 1.2 ms when the network is stable). The protocol which introduces minor delay (less than 1 ms.) and provides greatest stability is the OLSR. DSR is the protocol with worst features (in the 100 nodes topology its delay reaches the 2.33 ms. in the initial phase). The delay for all protocols is lower when we simulate the 250 nodes topology. It happens because the node has more neighbour nodes and more possible connections so the recovery time is lower.

Fig. 6 shows quite similar behaviour than in Fig. 5. The difference is not so high because of the mobility and errors. All cases have lower values than 1.1 ms when the network converges. OLSR is the protocol that introduces lower delay. It has a mean value of 0.91 ms. for case of 250 nodes. Reactive protocols have greater delay because they must know the network topology every time they establish a path between a source and a destination. DSR is the protocol which introduces more delay in the 100 nodes topology (2.9 ms. in the initial phase).

4.5. Measurements in a sensor node

Previous subsections analyzed the behaviour of the MANET protocols from the point of view of the network. Measurements were taken from the entire network. Now, we are going to show the same cases but from the point of view of the node. It will allow us to know which amount of that is able to transmit the sensor nodes with that features. It will also show us which the most suitable protocol to use in WSANs is.

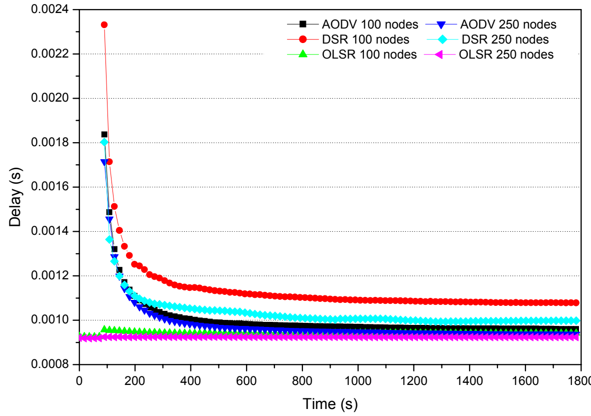


Fig. 5. Average delay in fixed nodes

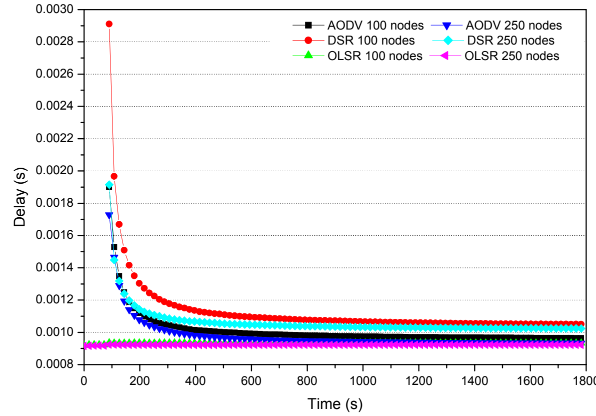


Fig. 6. Average delay in mobile nodes and failures

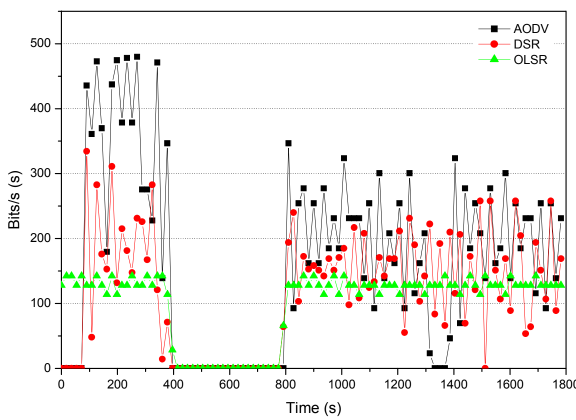


Fig. 7. Routing traffic in a mobile node

This time, we used the mobility and error scenario in the 250 nodes topology because of two main reasons: a) that topology behaviour is similar to the real WSANs, and b) that topology is the worst case.

Fig. 7 shows the processed routing traffic by the sensor node. We can observe that OLSR is the protocol that introduces less network traffic (150 bits/s). During the interval of 400 to 800 seconds there is not any routing traffic because in that time interval the node has failed. OLSR is the most stable protocol. DSR and AODV protocols have very strong fluctuations when the network is in the initial phase or recovering the node. These high peaks have unwanted effects on the sensor nodes because they imply greater energy consumption. Remember that WSAN protocols try to save energy in the sensor nodes.

Fig. 8 shows the average routing traffic in a sensor node. There can be seen better the features we have analyzed previously.

The average throughput highly depends on the routing protocol as it can be seen in Fig. 9. DSR is the one which consumes lower throughput rate (100 bit/s when the network converges), OLSR has a throughput rate consumption of 300 bits/s. We can see that all protocols have the same behaviour when an error occurs in the network. Around the 800 seconds all protocols fall to the half of its maximum value.

The delay is very low for all simulated protocols as in Fig. 10 can be seen. DSR protocol introduces the lowest delay when the network converges. OLSR is the most stable one. The delay of the three protocols fluctuates between 0.95 ms. and 0.75 ms. when the network converges.

In Fig. 11 the behaviour of a sensor node when it must send data is shown. Each protocol has different peaks, but the highest ones are from the DSR protocol. The AODV and OLSR behaviour are adequate, but the OLSR sends less quantity of bits in the same interval time. In WSAN, it is better to send few packets with some data that a single packet with very much data. This procedure saves energy giving larger time to life to the nodes.

5. Conclusions

MANET standard protocols were not designed to operate on WSANs, but we have demonstrated that, because they introduce very few traffic, they can be used in such networks.

All simulated protocols could be implemented in WSAN, the routing protocol which fits best with the WSAN characteristics is the OLSR protocol. Even when there are a large number of nodes with mobility and failures/recoveries, the traffic is stable and lower than in the other two simulated protocols. But, when we simulate the consumed throughput rate, the worst protocol is the OLSR because it is a proactive protocol.

All the routing protocols have an average delay close to zero for the simulated scenarios (the worst case varies between 1 and 3 milliseconds).

Currently, MANET working group is designing the OLSRv2, so further studies should test its performance.

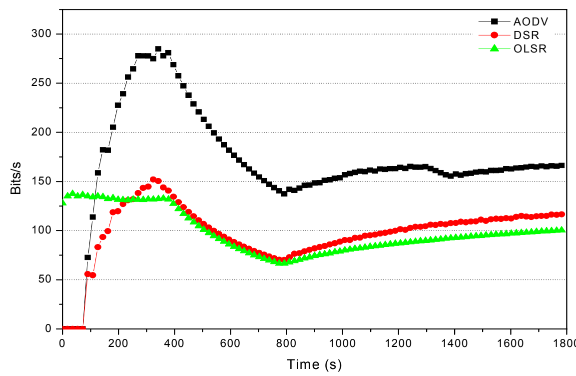


Fig. 8. Average routing traffic in a mobile node

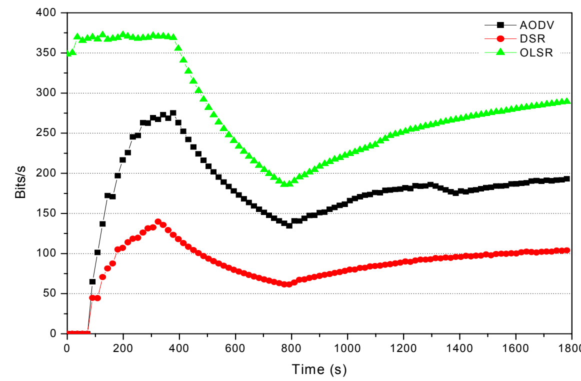


Fig. 9. Average throughput traffic in a mobile node

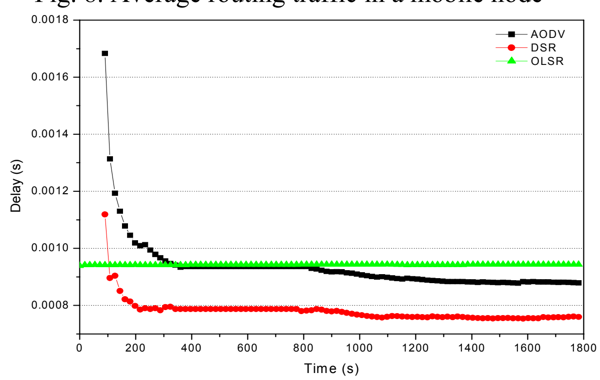


Fig. 10. Average delay traffic in a mobile node

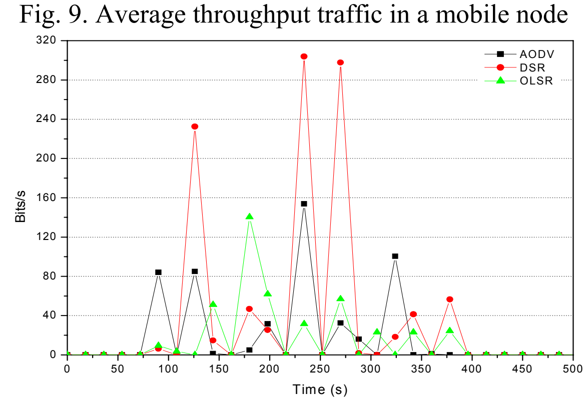


Fig. 11. Data traffic in a mobile node

6. References

[1] IETF Working Group Charter (1997) http://www.ietf.org/html/charters/manet-charter.html
[2] Ian F. Akyildiz, Ismail H. Kasimoglu, "Wireless sensor and actor networks: research challenges", Ad Hoc Networks, Volume 2, Issue 4. Pages 351-367 October 2004.
[3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks", MobiCom 99. Seattle, USA, August 1999.
[4] Callaway, E. H, "Wireless Sensor Networks: Architectures and Protocols". CRC Press, Inc. 2004
[5] R. Misra, C. R. Mandal, "Performance comparison of AODV/DSR on-demand routing protocols for ad hoc networks in constrained situation", IEEE Int. Conf. on Personal Wireless Communications. Pp. 86- 89. Jan. 2005.
[6] G. Jayakumar, G. Ganapathy, "Performance Comparison of Mobile Ad-hoc Network Routing Protocol", Int. J. of Comp. Science and Net. Security, Vol.7 No.11, Nov. 2007.
[7] H. Jiang and J. J. Garcia-Luna-Aceves, "Performance comparison of three routing protocols for ad hoc networks", Proceeding of IEEE ICCCN 2001.
[8] S. Ahmed and M. S. Alam, "Performance Evaluation of Important Ad Hoc Network Protocols". EURASIP Journal on Wireless Com. and Networking. Vol. 2006. Issue 2. Pp. 1-11.
[9] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks", 5th ACM/IEEE Int. Conf. on Mobile computing and networking. Pp. 195-206. 1999.
[10] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", Proc. of the 4th ACM/IEEE Int. conf. on Mobile computing and networking, Pp. 85-97. 1998.
[11] Azzedine Boukerche, "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks", Journal of Mobile Networks and Ap. Vol. 9, No 4. Pp.333-342, 2004.
[12] Al-Karaki J.N., Kamal A.E., "Routing techniques in wireless sensor networks: a survey", IEEE Wireless Communications, Vol. 11, Issue: 6, Pp: 6- 28, Dec. 2004.
[13] Polastre, J., Hill, J., and Culler, D. "Versatile low power media access for wireless sensor networks", Proc. of the 2nd international Conference on Embedded Networked Sensor Systems. Pp. 95-107, New York, 2004.
[14] T. Clausen and P. Jacquet. "Optimized Link State Routing Protocol (OLSR)". RFC 3626. October, 2003.
[15] Ogier, R., Templin, F., and Lewis, M. "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)". RFC 3684. February, 2004.
[16] C. Perkins, E. Belding-Royer and Das, S. "Ad Hoc On-Demand Distance Vector Routing". RFC 3561. July, 2003.
[17] D. Johnson, Y. Hu, D. Maltz. "The Dynamic Source Routing Protocol (DSR) for Mobile Ad-hoc Networks for IPv4". RFC 4728. February, 2007.
[18] ZJ. Haas, MR. Pearlman. "The zone routing protocol (ZRP) for ad hoc networks" (Internet-Draft), - Mobile Ad-hoc Network (MANET) Working Group, IETF, Aug, 1998
[19] OPNET Modeler website. Available at, http://www.opnet.com/solutions/network_rd/modeler.html

MANET Protocols Performance in Group-based Networks

Jaime Lloret[1], Miguel García[2], Fernando Boronat[3] and Jesus Tomás[4]

Department of Communications, Polytechnic University of Valencia

Camino Vera, s/n, Valencia 46022, Spain

[1]jlloret@dcom.upv.es, [2]migarpi@posgrado.upv.es, [3]fboronat@dcom.upv.es, [4]jtomas@upv.es

Abstract Many routing protocols for ad-hoc and sensor networks have been designed, but none of them are based on groups. It is known that grouping nodes gives better performance to the group and to the whole system, thereby avoiding unnecessary message forwarding and additional overhead. In this paper we show the efficiency of the MANET routing protocols when the nodes are arranged in groups. In order to do it, first, we study the advantages of grouping nodes in each individual protocol for both fixed and mobile networks (nodes with random mobile behaviour). Then, the routing protocols will be compared to analyse which one has the best performance when it is used in a group-based network. This paper shows that group-based systems applied to ad-hoc networks provides better performance than when they are not arranged in groups.

1 Introduction

MANET networks [1] are a type of ad-hoc networks much more studied and mature than Wireless Sensor Networks (WSN) [2]. Independently of the medium access method used [3], in the recent years many routing protocols have been developed for MANET networks [4] [5]. The nodes' mobility, the lack of stability of the topology, the lack of a pre-established organization and performing of the wireless communications are reasons for not using the routing protocols developed for fixed networks. There are standardized routing protocols for MANET networks used by different fixed or mobile devices.

Depending on the information type exchanged by the nodes and on the frequency they do it, the routing protocols in ad-hoc networks are divided into three types: proactives, reactives and hybrids. The proactive protocols update the routing tables of all the nodes periodically, even though no information is being exchanged. When a topology change occurs, the routing table is updated and the

Please use the following format when citing this chapter:

Lloret, J., García, M., Boronat, F. and Tomás, J., 2008, in IFIP International Federation for Information Processing, Volume 284; *Wireless and Mobile Networking*; Zoubir Mammeri; (Boston: Springer), pp. 161–172.

routing protocol finds the optimum route to forward the information. A periodical control protocol message exchange allows this, but consumes bandwidth and energy (batteries). The reactive protocols only maintain routing routes in their tables when a node has to communicate with another node in the network. With these protocols, when a communication starts, as the right route is unknown, a route discovering message is sent. When the response is received, the route is included in the routing tables and the communication is now possible. The main disadvantage of these protocols is the latency at the beginning of the communications (route discovery time) but they improve the network and energy resources use. Inside this kind of protocols, we can find the source-based protocols and hop-by-hop protocols. Finally, the hybrid protocols are a combination of the other two types, taking the advantages of both types. These protocols divide ad-hoc networks into different zones, and then near nodes use proactive routing meanwhile far nodes use reactive routing.

Current IETF standardized protocols are AODV (Ad-Hoc On-demand Distance Vector) [6], DSR (Dynamic Source Routing Protocol) [7] and OLSR (Optimized Link State Routing Protocol) [8]. So, we are going to analyze and study their performance in this paper.

AODV is a routing protocol for Mobile ad-hoc networks and is a reactive protocol. It has a minimalist behaviour because it hardly overloads the ad-hoc network and needs very few memory comparing with other protocols. It works over IP protocol.

DSR is a reactive protocol developed specifically for ad-hoc networks. It only sends information when it is required, saving bandwidth, energy and battery. The protocol has two mechanisms: route discovering and route maintenance. It also includes a mechanism to avoid loops. It is compatible with IPv6 (IP, version 6). It has the following disadvantages: the excessive initial latency while discovering the route; and, as a source-based protocol, the size of the packet header increases each time it goes through a node, affecting to the bandwidth consumption.

OLSR is a proactive protocol in which each node knows permanently the network state and the number, availability and addresses of the nodes. This performs a faster routing protocol. OLSR optimizes the time of response when a change is detected in the network, by reducing the period time of the control messages transmission. As routes for all the destinations are maintained, it has a quite good performance in networks with random and sporadic traffic in large groups of nodes. As disadvantages we can point the followings: the nodes require more memory resources and it overloads the network with routing control packets. OLSR was developed to work, in an independently way, with other protocols, bringing versatility to use it in any scenario. The most important key in this protocol is MPR (Multipoint Relay) node, it optimizes the number of control messages in the network.

There are several works published that compare MANET routing protocols [9] [10] [11] [12], but none of them have compared MANET routing protocols from the group-based network point of view. In order to do our comparison, we have used the version Modeler of OPNET simulator [13]. Our goal is to evaluate the performance of three MANET routing protocols from the point of view of some parameters such as: network load when the network is stable, network load when there are topology changes, convergence time, number of updates, correct sent/received packets, wrong sent/received packets, etc.

This paper is structured as follows. Section 2 explains group-based networks benefits, describes some group-based protocols and explains where group-based protocols could be implemented. Our simulations, analysis and comparison are shown in section 3. Finally, section 4 gives the conclusions and future works.

2 Group-based networks benefits and related works

A group is referred as a small number of interdependent nodes that interact in order to share resources or computation time and produce joint results. In a physical group-based architecture neighboring groups could be connected if border nodes from different groups are close. A group-based network is capable of having different types of topologies and protocols inside every group. There are some works in the literature where nodes are divided into groups and links between nodes from different groups are taken into account, but all of them have been developed to solve specific issues and none of them for MANET networks. Rhubarb system [14] organizes nodes in a virtual network, allowing connections across firewalls/NAT, and efficient broadcasting. Rhubarb system has only one coordinator per group and coordinators could be grouped in groups in a hierarchy. A Peer-to-Peer Based Multimedia Distribution Service was presented in [15]. Authors propose a topology-aware overlay in which nearby hosts or peers self-organize into application groups. End hosts within the same group have similar network conditions and can easily collaborate with each other to achieve QoS awareness. There are some architectures, such as [16] and [17], were nodes are structured hierarchically and parts of the tree are grouped into groups. In some cases, some nodes have connections with nodes from other groups although they are in different layers of the tree, but the information has to be routed through the hierarchy.

Finally, we want to emphasize that the cluster-based networks are a subset of the group-based networks, because every cluster could be considered as a group. A cluster can be made up of a Cluster Head node, Cluster Gateways and Cluster Members ([18] [19]). The Cluster Head node is the parent node of the cluster, which manages and checks the status of the links in the cluster, and routes the information to the right clusters. The rest of the nodes in a cluster are all cluster members and don’t use to have inter-cluster links. The size of the cluster is usually about 1 or 2 hops from the Cluster Head node while a group could be as large as we want. All the clusters have the same rules, however, a group-based network is capable of having any type of topology inside the group, not only clusters.

Group-based networks provide some benefits for the whole network such as:

- Spreads the work to the network in groups giving more flexibly, efficiently and lower delays.
- Content availability will increase because it could be replicated to other groups.
- Anyone could search and download from every group using only one service.
- It provides fault tolerance. Other groups could carry out tasks from a failed one.
- It is scalable because a new group could be added to the system easily.
- Network measurements could be taken from any group.

Group-based networks have many application areas. They could be used when it is wanted to setup a network where groups could appear and join the network anytime or by networks have to be split into smaller zones to support a large number of nodes, that is, any system where the devices are grouped and there must be connections between groups. The main goal in a group-based topology is the network protocol and the group management, that is, the design of an efficient algorithm for a new node to find its nearest (or the best) group to join in. The performance of the network highly depends on the efficiency of the nearby group locating process and on the interaction between neighboring groups.

The following list gives several group-based ad-hoc networks application areas:

1. Let us suppose a job where all human resources need to be split into groups to achieve a purpose (such as fire fighter squads for putting out the fire). Now, let's suppose that all people involved in that activity need a device that has to be connected with other devices in the same group to receive information from the members within the group, and closer groups have to be connected to coordinate their efforts. Currently coordination between groups is done through a wireless connection to the command center or using satellite communications. But, some times neither of those solutions can be used because a free of obstacles line of sight is needed, because there are too many wall looses or because more gain or power is needed to reach the destination.
2. Groups could also be established because of geographical locations or unevenness. It happens in rural and agricultural environments. A group-based topology in this kind of environment could be useful to detect plagues or fire and to propagate an alarm to neighbor lands. It will provide easier management and control for detecting fires and plagues as well as allowing scalability.
3. It could be used in any kind of system in which an event or alarm is based on what is happening in a specific zone, but conditioned to the events that are happening in neighbor zones. One example is a group-based system to measure the environmental impact of a place. It could be better measured if the measurements are taken from different groups of nodes, but those groups of nodes have to be connected in order to estimate the whole environmental impact.
4. Group-based virtual games. There are many games where the players are grouped virtually in order to perform a specific task. Interactions between groups in virtual reality should be given by interactions between players from different groups to exchange their knowledge.

3 Group-based ad-hoc networks analysis

This section describes how are the test bench and the traffic used for simulations, in the regular and the group-based topologies, to take measurements.

3.1 Test bench

We used the same test-bench for all the evaluated protocols using OPNET Modeler [13]. We varied the number of nodes and the coverage area in an open environment. The nodes in the topology have the characteristics of an ad-hoc node (40 MHz processor, 512 KB memory card, radio channel of 11 Mbps and 2.4 GHz as the work frequency). The MAC protocol was CSMA/CA. We chose nodes with a 50 meters maximum coverage radius. This is a conservative value, so the simulations presented in this work give us an adequate view for the worst case.

We simulated 4 scenarios for each protocol: the first one with fixed nodes; the second one with mobile nodes and failures; the third one with grouped nodes; and, the fourth one with grouped mobile nodes and failures. For each topology, we simulated for 100 and 250 nodes, to observe the system scalability.

Instead of a standard structure we chose a random topology where the nodes are mobile and change their position constantly. The groups are created by physical coverage area. When a node moves into a new coverage area, it belongs to the new group. 100 nodes topology has a 750x750 m^2 area (density ! $0.18\cdot10^{-3}$ nodes/m^2) and 250 nodes topology has a 1 Km^2 area (density ! $0.25\cdot10^{-3}$ nodes/m^2). We forced failures at t=200 sec., t=400 sec. and t=1200 sec. in each network, with a recovering process of 300 sec., to take measurements when the physical topology changes. We are going to study how several network-level protocols perform when failures and recoveries happen in this kind of networks.

We created 6 groups for the 100 nodes topology, each group covers a circular area of 150 meter radius. They were arranged to cover the whole area. There were approximately 16 or 17 nodes, in each group in the initial process. The number of nodes in each group varied because of the node's random mobility, so in one instant a node could belong to a group and in another instant to another one. We created 12 groups with 15 or 16 nodes per group for the 250 nodes topology. The group's coverage areas were similar for both areas. The routing protocol used inside the group will be the same as the one used between groups.

The traffic load used for the simulations is MANET traffic generated by OPNET. We inject this traffic 100 seconds after the beginning. The traffic follows a Poisson distribution (for the arrivals) with a mean time between arrivals of 30 seconds. The packet size follows an exponential distribution with a mean value of 1024 bits. The injected traffic has a random destination address, to obtain a simulation independent of the traffic direction. We have simulated the four scenarios for DSR, AODV and OLSR protocols.

3.2 Simulation results and analysis for DSR protocol

In figures 1 and 2 we can see the MAC level mean delay experimented by traffic using CSMA/CA. In figure 1, the group-based topology has a mean delay of 250 μs independently of the number of nodes in the network. The regular topology converges around 1.1 ms. The difference between both cases is about 850 μs; therefore the MAC level mean delay decreases a percentage of 23% in both cases. In figure 2, we can see that the delays are lower. It is mainly because of the network mobility. In this case, we appreciate that there are differences between 100-node and 250-node topologies. In group-based topologies the MAC level mean delay is around 100 μs for both topologies, so they converge faster.

When we study the network throughput consumed (figures 3 and 4), we observe that group-based topologies give a much lower value than the one obtained in regular topologies. For the 100-node topology (figure 3), the mean throughput varies from 225 Kbits/s to 100 Kbits/s in the group-based topology (a 56% improvement). In the 250-node topology we obtain 460 Kbits/s of throughput for the regular topology and 190 Kbits/s of throughput for the group-based one (a 59% improvement). Moreover, when we compare figures 3 and 4, we can conclude that the throughput in group-based topologies has a very low variation regarding a fixed or mobile scenario. The obtained improvement is quite important. We can see in figure 4 that, after 1200 seconds, the obtained throughput in 250-node topology is similar to the obtained throughput in the 100-node regular topology.

Observing figures 5 and 6, we conclude that the MANET traffic load through the network is lower for group-based topologies. In both figures we can see that when the number of nodes increases the traffic decreases. This is due to the existence of more nodes working as routers and therefore the probability of a packet to reach the destination is higher. The 100-node group-based topology (figure 5) gives a 77% improvement regarding the regular 100-node topology, but the improvement decreases when the number of nodes increases (in 250-node cases the improvement is about 60%). This behavior varies when the topology has mobility, errors and failures (figure 6). In this case, the 100 node group-based topology also has improvement (around 77%) regarding to the 100-node regular one. This improvement is higher (80%) in 250-node topologies.

We have also compared the routing traffic sent (figures 7 and 8). In figure 7 we observe that the traffic is quite stable due to the characteristics of the network, because it is a fixed network without errors and failures. The traffic sent in 250-node topology is around 225 Kbits/s, but when we group the nodes this traffic decreases to 100 Kbits/s (a 60% improvement). The value obtained in a 100-node topology is also improved when we group the nodes (50 Kbits/s, therefore a 50% improvement). In figure 8 we observe a similar behavior. In this case we conclude that when there are errors and failures in the network (interval from 600 to 800 seconds and around 1200 seconds) the traffic fluctuates and is less stable. We appreciate the instability is much lower in group-based topologies.

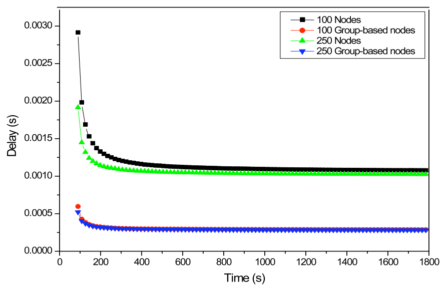


Fig. 1 DSR mean delay in fixed topologies.

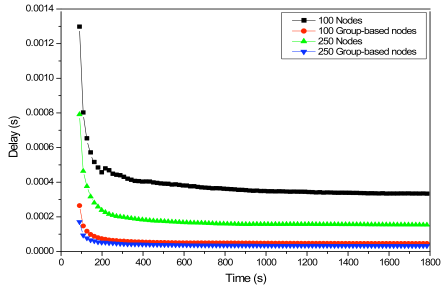


Fig. 2 DSR mean delay in mobile topologies.

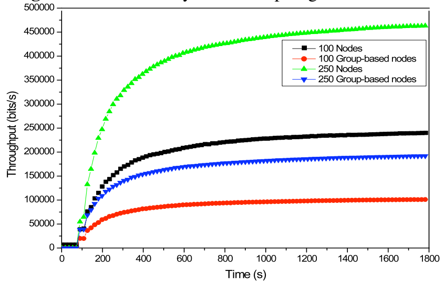


Fig. 3 DSR mean throughput in fixed topologies.

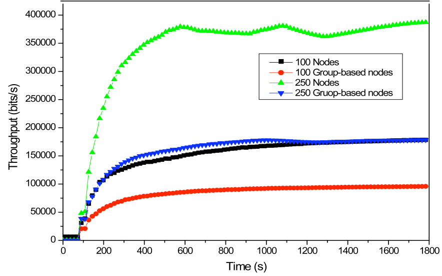


Fig. 4 DSR mean throughput in mobile topologies.

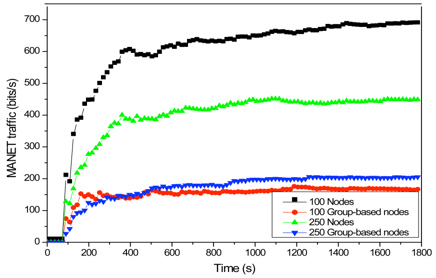


Fig. 5 DSR mean MANET traffic in fixed topologies.

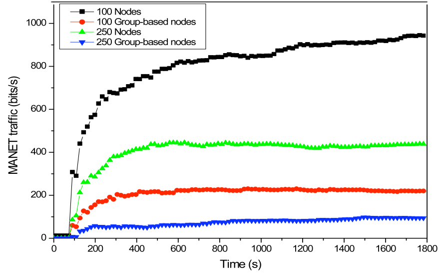


Fig. 6 DSR mean MANET traffic in mobile topologies.

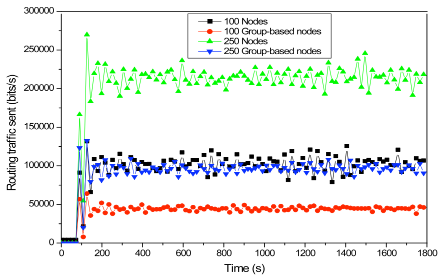


Fig. 7 DSR routing traffic in fixed topologies.

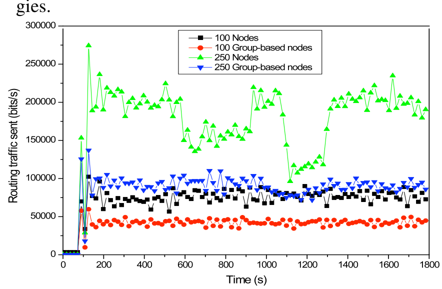


Fig. 8 DSR routing traffic mobile topologies.

3.3 Simulation results and analysis for AODV protocol

Figures 9 and 10 show the MAC level mean delay for AODV protocol simulations. We can observe that the delay has no strong dependence of both the topology type and the number of nodes. For 100 and 200-node topologies we obtain a delay stabilized around 1 ms, but for group-based topologies this value is 300 μs. Therefore it decreases a percentage of 70% in both cases.

Figure 11 shows the mean throughput for fixed topologies. The 100-node scenario gives a 200 Kbits/s mean value, but a value of 120 Kbits/s is obtained for the group-based scenario (a 40% improvement). In the 250-node case, we obtain mean values of 425 Kbits/s for the fixed scenario and of 225 Kbits/s for the group-based scenario (a 47% improvement). Figure 12 shows the results for mobile topologies with errors and failures. The improvement, obtained by grouping nodes, decreases in the 100-node case (37%), but it doesn't vary in the 250-node cases.

Figures 13 and 14 show the evolution of the MANET traffic for different scenarios. In figure 13, the traffic of the 100-node fixed topology has a mean value of around 600 bits/s, but it decreases to 180 bits/s for the 100-node group-based scenario, giving a 70% of improvement. In the 250-node topologies, it has varied from 480 Kbits/s (fixed) to 50 Kbits/s (group-based), obtaining a 90% improvement. In figure 14 we appreciate the improvement (not too relevant as in above case) and we can see the fast convergence of group-based topologies for mobile topologies with errors and failures. In the 100-node case the traffic doesn't converge before 1400 seconds, but it converges in 200 seconds when there are group of nodes. It also happens in the 250-node topologies: the regular topology converges in 600 seconds while the group-based one converges in 180 seconds.

The routing traffic measured in each simulated scenario can be seen in figures 15 and 16. We observe that the routing traffic is independent of the node mobility. In figure 15 we can see that the routing traffic goes from 200 Kbits/s for 250-node case to 125 Kbits/s when there are group of nodes (a 37% improvement). In the 100-node cases, it goes from 90 Kbits/s to 50 Kbits/s (a 45% improvement). When there are mobility, errors and failures (see figure 16), in the 250-node topology the values go from 190 Kbits/s to 110 Kbits/s in the group-based scenario (a 42% improvement). We obtained 85 Kbits/s for the regular 100-node topology and 50 Kbits/s for the group-based one (a 41% improvement).

3.4 Simulation results and analysis for OLSR protocol

Figure 17 shows the MAC level mean delay in fixed topologies. In the 250-node regular topology we obtained a value around 920 μs and a value around 250 μs in the group-based one (a 73% improvement). With 100 nodes there is no a significant improvement in the group-based topology (both values are around 260 μs).

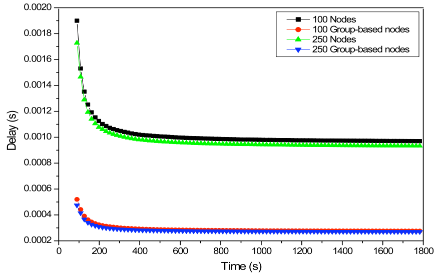


Fig. 9 AODV mean delay in fixed topologies

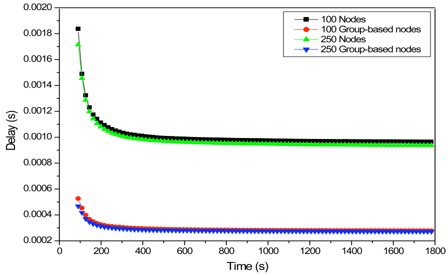


Fig. 10 AODV mean delay in mobile topologies.

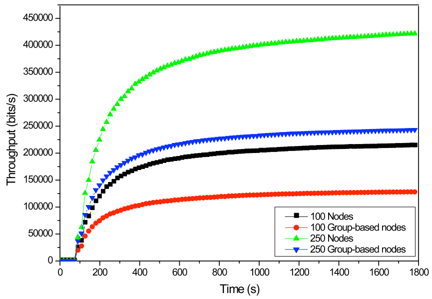


Fig. 11 AODV mean throughput consumed in fixed topologies.

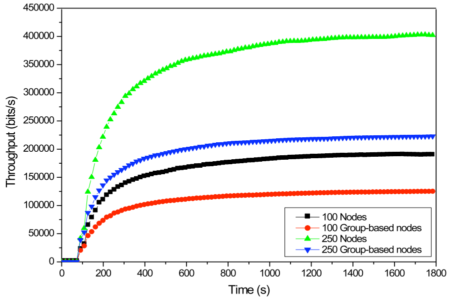


Fig. 12 AODV mean throughput consumed in mobile topologies.

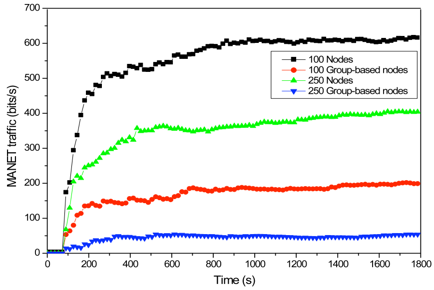


Fig. 13 AODV mean MANET traffic in fixed topologies.

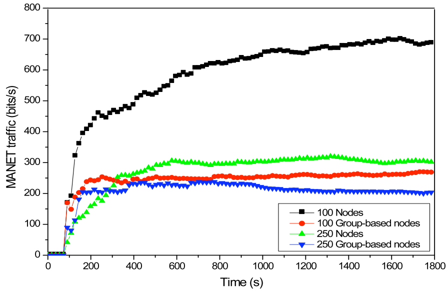


Fig. 14 AODV mean MANET traffic in mobile topologies.

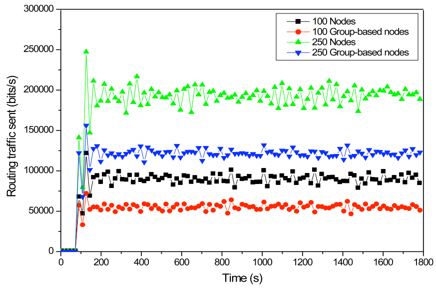


Fig. 15 AODV routing traffic in fixed topologies.

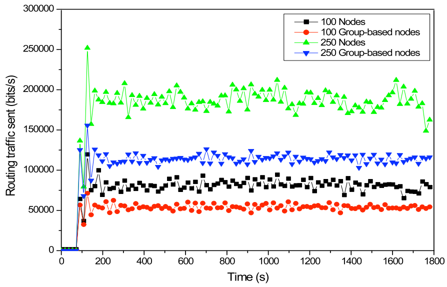


Fig. 16 AODV routing traffic in mobile topologies.

Figure 18 shows the cases when there is mobility, errors and failures. In 100-node topologies, the regular topology has a mean value of 268 μs and the group-based scenario has 262 μs. In 250-node case, the improvement is not so good (from 262 μs to 260 μs).

The mean throughput consumed in fixed topologies can be observed in figure 19. In the scenarios with 250 nodes we obtained throughputs of 550 Kbits/s and 250 Kbits/s (group-based has 54% improvement). In 100-node regular topology the throughput is 325 Kbits/s and 125 Kbits/s (group-based has 61% improvement). When we consider mobility (figure 20) the consumed throughput is not so stable as in above case but, we can observe that the improvements are quite similar. In the case of 250 nodes we obtain a 52% improvement in group-based scenario; in the case of 100 nodes the improvement reaches the 60%.

When we analyze the mean MANET traffic through the network, we obtain the results shown in figure 21 and 22. In figure 21, the mean traffic in the 100-node topology was 700 bits/s for the regular topology and 180 bits/s for the group-based topology, obtaining a 75% improvement. In 250-node regular scenario, we obtained around 450 bits/s, but a value of 220 bits/s when we group nodes. There is around 51% improvement. Figure 22 shows the value of MANET traffic in mobile topologies with errors and failures. We appreciate that regular topologies have higher convergence time than the group-based topologies. In this case, the 100-node topology did not converge in the simulated interval. In the 250-node topology, the network gets stability after 1200 seconds. The improvement of the 250-node topologies is around 51%. The mean value of the MANET traffic obtained in the regular 100-node topology is around 900 bits/s and 215 bits/s in the group-based topology, giving a 76% improvement.

Finally, we have analyzed the mean routing traffic sent in fixed and mobile topologies (figures 23 and 24). The routing traffic sent in the 100-node fixed topology was around 60 Kbits/s, while in the group-based topology was 28 Kbits/s, with a 53% improvement. In the 250-node case, we appreciate that this traffic was higher than 110 Kbits/s, but only higher than 60 Kbits/s for the group-based scenario, with a 45% improvement (figure 23). Figure 24 shows the results of a network with mobility, errors and failures. The routing traffic is quite dependent of the failures in the network. In both 100-node and 250-node scenarios there are some fluctuations due to the inherent characteristics of the network that are minimized grouping the nodes. Improvements of 45% and 53% are obtained in 100-node and 250-node scenarios, respectively. So, we can emphasize the good scalability of the group-based topologies.

4 Conclusions

In this work we have shown the benefits of using a group-based topology in ad-hoc networks and we have shown several examples where they can be used.

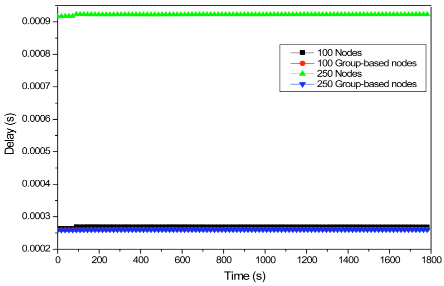


Fig. 17 OLSR mean delay in fixed topologies.

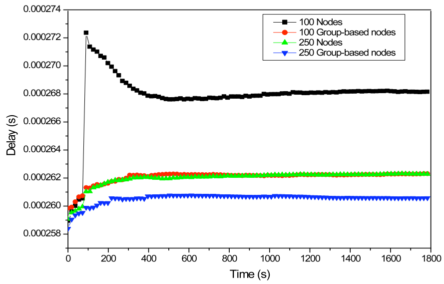


Fig. 18 OLSR mean delay in mobile topologies.

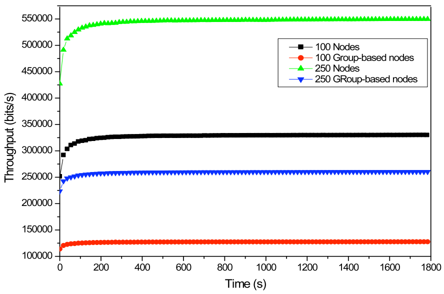


Fig. 19 OLSR mean throughput consumed in fixed topologies.

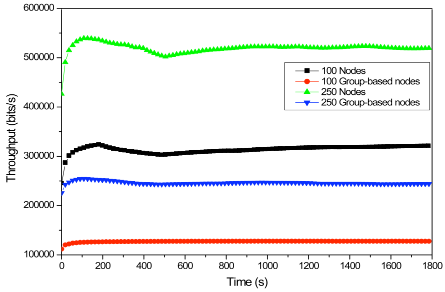


Fig. 20 OLSR mean throughput consumed in mobile topologies.

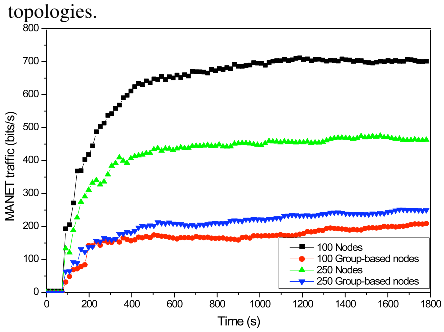


Fig. 21 OLSR mean MANET traffic in fixed topologies.

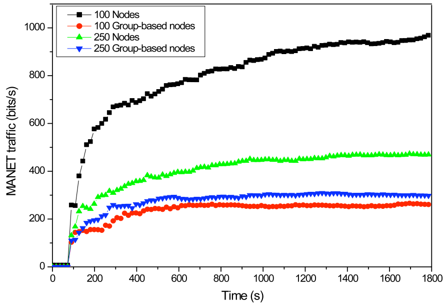


Fig. 22 OLSR mean MANET traffic in mobile topologies.

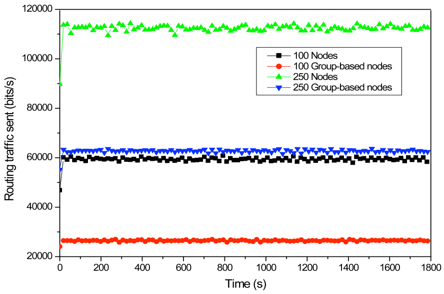


Fig. 23 OLSR routing traffic in fixed topologies.

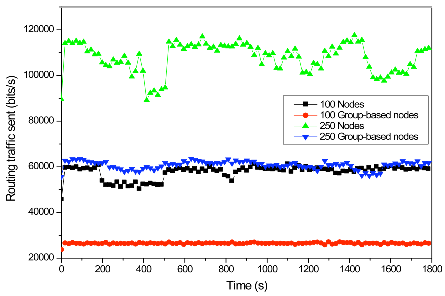


Fig. 24 OLSR routing traffic in mobile topologies.

We have simulated DSR, AODV and OLSR protocols with and without groups and the results show that group-based topologies give better performance for wireless ad-hoc networks. So, grouping nodes increases the productivity and the performance of the network with low overhead and low extra network traffic. Therefore, good scalability can be achieved in group-based networks. On the other hand, the protocol that gives better results has been OLSR because this protocol introduces less routing traffic and it has behavior more regular.

References

1. M. Frodigh, P. Johansson, P. Larsson. "Wireless ad hoc networking. The art of networking without a network". Ericsson Review, No. 4, 2000.
2. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. “A survey on sensor networks”. Communications Magazine, IEEE. Volume 40, Issue 8. Pp. 102- 114. Aug 2002.
3. S. Kumar, V. S. Raghavan, J. Deng. "Medium Access Control protocols for ad hoc wireless networks: A survey". Ad Hoc Networks. Volume 4, Issue 3, Pp. 326-358. May 2006.
4. E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communications, April 1999, pp. 46-55.
5. R. Rajaraman, “Topology control and routing in ad hoc networks: a survey”. ACM SIGACT News. Volume 33, Issue 2, Pp. 60-73. June 2002.
6. C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing". RFC 3561. July, 2003.
7. D. Johnson, Y. Hu, D. Maltz. "The Dynamic Source Routing Protocol (DSR) for Mobile Ad-hoc Networks for IPv4". RFC 4728. February, 2007.
8. T. Clausen and P. Jacquet. "Optimized Link State Routing Protocol". RFC 3626. Oct. 2003.
9. Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols". Proc. of the 4th ACM/IEEE int. conf. on Mobile computing and networking. Pp. 85-97. 1998.
10. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-Based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks”. Proc. of the 5th ACM/IEEE int. conf. on Mobile computing and networking. Pp. 195-206. 1999.
11. C. E. Perkins, E. M. Royer, S. R. Das, M. K. Marina, Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks, IEEE Personal Com. Pp. 16-28. Feb. 2001.
12. X. Hong, K. Xu, M. Gerla. "Scalable routing protocols for mobile ad hoc networks". IEEE Network. Volume 16, Issue 4, Pp. 11-21. Jul/Aug 2002.
13. OPNET Modeler website. At, http://www.opnet.com/solutions/network_rd/modeler.html
14. Wierzbicki, A., Strzelecki, R., Swierczewski, D. and Znojek, M. Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications. Second IEEE International Conference on Peer-to-Peer Computing (P2P2002), Linöping, Sweden, 2002.
15. Xiang, Z., Zhang, Q., Zhu, W., Zhang, Z. and Zhang, Y. Peer-to-Peer Based Multimedia Distribution Service, IEEE Transactions on Multimedia 6 (2) (2004).
16. L. Hongjun, L. P. Luo, Z. Zhifeng. A structured hierarchical P2P model based on a rigorous binary tree code algorithm, Future Generation Computer Systems 23 (2). 2007. Pp. 201-208.
17. B. Thallner, H. Moser. Topology control for fault-tolerant communication in highly dynamic wireless networks. 3rd Wrk. on Intelligent Solutions in Embedded Systems. May 2005.
18. M. Jiang, J. Li, Y.C. Tay. “Cluster Based Routing Protocol (CBRP)”. Internet-draf, draft-ietf-manet-cbrp-spec-01.txt, National University of Singapore, 14 August 1999.
19. Chiang, C.C., Wu, H.K., Liu, W., Gerla, M., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", IEEE SICON, Kent Ridge (Singapore). April 1997.