Universitat Politècnica de València

Departamento de Comunicaciones

Ph. D. Thesis

Design and Implementation of a Communication Protocol to Improve Multimedia QoS and QoE in Wireless Ad Hoc Networks

---

Presented by:

Juan Ramón Díaz Santos

Supervised by:

Jaime Lloret Mauri

Valencia, January 2016

Abstract

This dissertation addresses the problem of multimedia delivery over multi-hop ad hoc wireless networks, and especially over wireless sensor networks. Due to their characteristics of low power consumption, low processing capacity and low memory capacity, they have major difficulties in achieving optimal quality levels demanded by end users in such communications.

In the first part of this work, it has been carried out a study to determine the behavior of a variety of multimedia streams and how they are affected by the network conditions when they are transmitted over topologies formed by devices of different technologies in multi hop wireless ad hoc mode. To achieve this goal, we have performed experimental tests using a test bench, which combine the main codecs used in audio and video streaming over IP networks with different sound and video captures representing the characteristic patterns of multimedia services such as phone calls, video communications, IPTV and video on demand (VOD). With the information gathered in the laboratory, we have been able to establish the correlation between the induced changes in the physical and logical topology and the network parameters that measure the quality of service (QoS) of a multimedia transmission, such as latency, jitter or packet loss. At this stage of the investigation, a study was performed to determine the state of the art of the proposed protocols, algorithms, and practical implementations that have been explicitly developed to optimize the multimedia transmission over wireless ad hoc networks, especially in ad hoc networks using clusters of nodes distributed over a geographic area and wireless sensor networks.

Next step of this research was the development of an algorithm focused on the logical organization of clusters formed by nodes capable of adapting to the circumstances of real-time traffic. The stated goal was to achieve the maximum utilization of the resources offered by the set of nodes that forms the network,

allowing simultaneously sending reliably and efficiently all types of content through them, and mixing conventional IP data traffic with multimedia traffic with stringent QoS and QoE requirements. Using the information gathered in the previous phase, we have developed a network architecture that improves overall network performance and multimedia streaming. In parallel, it has been designed and programmed a communication protocol that allows implementing the proposal and testing its operation on real network infrastructures.

In the last phase of this thesis we have focused our work on sending multimedia in wireless sensor networks (WSN). Based on the above results, we have adapted both the architecture and the communication protocol for this particular type of network, whose use has been growing hugely in recent years.

Resumen

Esta tesis doctoral aborda el problema de la distribución de contenidos multimedia a través de redes inalámbricas *ad hoc* multisalto, especialmente las redes inalámbricas de sensores que, debido a sus características de bajo consumo energético, baja capacidad de procesamiento y baja capacidad de memoria, plantean grandes dificultades para alcanzar los niveles de calidad óptimos que exigen los usuarios finales en dicho tipo de comunicaciones.

En la primera parte de este trabajo se ha llevado a cabo un estudio para determinar el comportamiento de una gran variedad de flujos multimedia y como se ven afectados por las condiciones de la red cuando son transmitidos a través topologías formadas por dispositivos de diferentes tecnologías que se comunican en modo *ad hoc* multisalto inalámbrico. Para ello, se han realizado pruebas experimentales sobre una maqueta de laboratorio, combinando los principales códecs empleados en la transmisión de audio y video a través de redes IP con diversas capturas de sonido y video que representan patrones característicos de servicios multimedia tales como las llamadas telefónicas, videoconferencias, IPTV o video bajo demanda (VOD). Con la información reunida en el laboratorio se ha podido establecer la correlación entre los cambios inducidos en la topología física y lógica de la red con los parámetros que miden la calidad de servicio (QoS) de una transmisión multimedia, tales como la latencia el jitter o la pérdida de paquetes. En esta fase de la investigación se realiza un estudio para determinar el estado del arte de las propuestas de desarrollo e implementación de protocolos y algoritmos que se han generado de forma explícita para optimizar la transmisión de tráfico multimedia sobre redes ad hoc inalámbricas, especialmente en las redes inalámbricas de sensores y redes ad hoc utilizando clústeres de nodos distribuidos en un espacio geográfico.

El siguiente paso en la investigación ha consistido en el desarrollo de un algoritmo propio para la organización lógica de clústeres formados por nodos capaces de adaptarse a las circunstancias del tráfico en tiempo real. El objetivo planteado es conseguir un aprovechamiento máximo de los recursos ofrecidos por el conjunto de nodos que forman la red, permitiendo de forma simultánea el envío de todo tipo de contenidos a través de ellos de forma confiable y eficiente, permitiendo la convivencia de tráfico de datos IP convencional con tráfico multimedia con requisitos exigentes de QoS y QoE. A partir de la información conseguida en la fase anterior, se ha desarrollado una arquitectura de red que mejora el rendimiento general de la red y el de las transmisiones multimedia de audio y video en particular. De forma paralela, se ha diseñado y programado un protocolo de comunicación que permite implementar el modelo y testear su funcionamiento sobre infraestructuras de red reales.

En la última fase de esta tesis se ha dirigido la atención hacia la transmisión multimedia en las redes de sensores inalámbricos (WSN). Partiendo de los resultados anteriores, se ha adaptado tanto la arquitectura como el protocolo de comunicaciones para este tipo concreto de red, cuyo uso se ha extendido en los últimos años de forma considerable.

Resum

Esta tesi doctoral aborda el problema de la distribució de continguts multimèdia a través de xarxes sense fil ad hoc multi salt, especialment les xarxes sense fil de sensors que, a causa de les seues característiques de baix consum energètic, baixa capacitat de processament i baixa capacitat de memòria, plantegen grans dificultats per a aconseguir els nivells de qualitat òptims que exigixen els usuaris finals en eixos tipus de comunicacions.

En la primera part d'este treball s'ha dut a terme un estudi per a determinar el comportament d'una gran varietat de fluxos multimèdia i com es veuen afectats per les condicions de la xarxa quan són transmesos a través topologies formades per dispositius de diferents tecnologies que es comuniquen en mode ad hoc multi salt sense fil. Per a això, s'han realitzat proves experimentals sobre una maqueta de laboratori, combinant els principals códecs empleats en la transmissió d'àudio i vídeo a través de xarxes IP amb diverses captures de so i vídeo que representen patrons característics de serveis multimèdia com son les cridades telefòniques, videoconferències, IPTV o vídeo baix demanda (VOD). Amb la informació reunida en el laboratori s'ha pogut establir la correlació entre els canvis induïts en la topologia física i lògica de la xarxa amb els paràmetres que mesuren la qualitat de servei (QoS) d'una transmissió multimèdia, com la latència el jitter o la pèrdua de paquets. En esta fase de la investigació es realitza un estudi per a determinar l'estat de l'art de les propostes de desenvolupament i implementació de protocols i algoritmes que s'han generat de forma explícita per a optimitzar la transmissió de tràfic multimèdia sobre xarxes ad hoc sense fil, especialment en les xarxes sense fil de sensors and xarxes ad hoc utilitzant clusters de nodes distribuïts en un espai geogràfic.

El següent pas en la investigació ha consistit en el desenvolupament d'un algoritme propi per a l'organització lògica de clusters formats per nodes capaços

d'adaptar-se a les circumstàncies del tràfic en temps real. L'objectiu plantejat és aconseguir un aprofitament màxim dels recursos oferits pel conjunt de nodes que formen la xarxa, permetent de forma simultània l'enviament de qualsevol tipus de continguts a través d'ells de forma confiable i eficient, permetent la convivència de tràfic de dades IP convencional amb tràfic multimèdia amb requisits exigents de QoS i QoE. A partir de la informació aconseguida en la fase anterior, s'ha desenvolupat una arquitectura de xarxa que millora el rendiment general de la xarxa i el de les transmissions multimèdia d'àudio i vídeo en particular. De forma paral·lela, s'ha dissenyat i programat un protocol de comunicació que permet implementar el model i testejar el seu funcionament sobre infraestructures de xarxa reals.

En l'última fase d'esta tesi s'ha dirigit l'atenció cap a la transmissió multimèdia en les xarxes de sensors sense fil (WSN). Partint dels resultats anteriors, s'ha adaptat tant l'arquitectura com el protocol de comunicacions per a aquest tipus concret de xarxa, l'ús del qual s'ha estés en els últims anys de forma considerable.

Agradecimientos

Con la presentación de esta tesis culmina un proceso de aprendizaje profesional y personal vinculado a la investigación científica. Este no es el final del camino, sino solamente una parada para recapitular el camino andado y seguir adelante. No me hubiera sido posible llegar hasta aquí sin la ayuda y el apoyo constante de mi director de tesis, Jaime Lloret, al que quiero agradecer el tiempo y el esfuerzo dedicado en la revisión de cada una de las partes que han formado esta investigación y, sobretodo, por la energía y el entusiasmo con la que aborda su trabajo y que me ha transmitido durante este tiempo.

Desde que hace varios años comenzará con las primeras pruebas experimentales que servirían de base para la investigación posterior, me he enriquecido con los conocimientos y habilidades de muchos compañeros. La aportación que han realizado a esta tesis José Miguel Jiménez, Sandra Sendra, Miguel García Pineda, Diana Bri, Mohammed Hammoummi o Fernando Boronat es inestimable para mí. Desde la ayuda prestada en el diseño y puesta en marcha de los procedimientos experimentales, pasando por la revisión de los modelos y protocolos utilizados, hasta las orientaciones y ayudas ofrecidas para realizar la comunicación de los resultados a través de artículos. Todos ellos han hecho posible que el producto de esta investigación se haya publicado en varias de las revistas científicas internaciones con mayor impacto dentro del campo de investigación de las nuevas tecnologías y las telecomunicaciones.

No puedo olvidar en este apartado de agradecimientos a Joel Rodrigues y a su equipo de trabajo, en la universidad de Beira Interior de Covilha, que durante mi estancia en Portugal me acogieron y me ayudaron a expandir mis puntos de vista, adquiriendo nuevas técnicas y conocimientos.

Por último, quiero agradecer de forma muy especial y entrañable a mi familia que me ha permitido alcanzar el equilibrio entre la faceta personal y profesional de la

vida, empujándome con su confianza a seguir avanzando y a poder disfrutar con mi trabajo.

He contado con todos vosotros cuando lo he necesitado, podéis contar conmigo cuando lo necesitéis.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction and Objectives

1.1 Introduction

The standardization of wireless technologies and the technological advances have enabled the fabrication of ever smaller, cheaper and more powerful devices. They have meant that, nowadays, wireless technology has become very popular and we can find it incorporated in virtually any electronic device. Smartphones, sensors and every kind of devices for personal or household use, such as watches or appliances include a wireless interface to communicate with other networked devices. We are moving towards a global network where people and devices share information.

Against this background of high wireless connectivity, new opportunities and new challenges show up. In particular, inside the wireless technology family, there has been a great proliferation of ad hoc wireless networks. This kind of wireless networks are built by some number of wireless devices scattered over a limited geographical area. These devices can use specific technologies and applications [1] [2]. They offer several advantages for a wide range of applications due to their ease of deployment, adaptability to environment, coverage control, mobility of the devices, consumption restrictions and energy saving [3]. Applications of such networks are *Wireless Sensor Network (WSN)* [4]*, Vehicular Ad Hoc Network (VANET)* [5] *Vehicular Ad Hoc and Sensor Network (VASNET)* [6] *Mobile Ad Hoc Network (MANET)* [7] *and Wireless Mesh Network (WMN)* [8].

Regardless of the original purpose for which they have been designed and implemented, these network infrastructures can provide us a wealth of resources that are often not used, for example the sensor networks usually only consume a reduced portion of available bandwidth and the rest remains unused. If the availability of resources in ad hoc wireless networks is considered together with the more and more user services demand, such as multimedia traffic, it leads to research about a possible solution that allows organizing and managing ad hoc networks for transmission of audio and video, in order to increase the range and availability of such kind of services: IP telephony (VoIP), video conferencing, IP television (IPTV) and video on demand (VoD). This Ph.D. thesis is focused on studying wireless ad hoc networks features in order to address the problem of guaranteeing the quality of service when transmitting multimedia traffic.

In this dissertation, we design and develop a generic architecture for the implementation of resource reservation and management of quality of service in wireless ad hoc networks. This model enables these network infrastructures for the spread of a wide range of multimedia services. Then, the proposed system is used as reference for designing a communication protocol that implements the features of the exposed architecture. The protocol is aimed at optimizing multimedia network traffic using characteristics from each specific network technology, but, particularly focusing on wireless ad hoc network. The research was completed with architecture and protocol validation by implementing a software application that can be used both on laboratory models and over real network scenarios.

The scientific contributions and the main results from the research conducted in this Ph.D. thesis have been published on several international journals with outstanding impact factor, which are listed in the conclusion section.

1.2 Objectives

From the beginning to the final results, several secondary goals have been achieved, but there has been a main objective present in each stage of the research process:

- **Propose a new architecture and protocol to improve the user quality of experience in the transmission of multimedia content over wireless ad hoc networks.**

  Nowadays we can see the proliferation of all kind of electronic devices such as mobile phones, tablets, access points or sensors that have the ability to connect to the network through wireless technologies. When they are considered as a whole, they are a valuable source of resources for deploying multimedia services such as voice calls, video conferencing, IPTV and video on demand and deliver them to anywhere in anytime.

However, due to the characteristics of wireless technologies and the high level of requirements for this type of transmission, it is mandatory to add management elements, smart software components and new protocols to build and manage appropriate communication channels.

To achieve this ambitious main objective, a work plan was developed including more specific goals in order to build a step by step solution. Following the chronological order that they were addressed, these objectives are:

- **Determine how wireless technologies can affect at the user perceived quality of streaming of audio and video.**

  This objective is looking for a definition of the problem we are facing us. The starting point for any investigation should always be to gather the current knowledge about the studied matter. To achieve this objective, on the one hand the scientific literature is revised and the relevant publications in this field are collected and analyzed to set the state of the art, and on the other hand, empirical research under controlled laboratory conditions must also be performed. There are a lot of magnitudes and variables that can be considered on an infrastructure wireless ad hoc network, such as interference between devices, electromagnetic noise, device density, traffic pattern on the network, device characteristics, mobility or logical organization of devices. We want to learn about which are the most relevant and the way we can control their influence.

- **Study how the changes on the logical topology in multi-hop wireless ad-hoc networks can benefit or harm multimedia traffic.**

  Several proposals about how organize the devices in an ad-hoc network, whether wired or wireless network, can be found in the scientific literature. These proposals are mainly thought for general purposes and for any kind and pattern of traffic but they are not specific for multimedia communications. For this reason, with this goal, we focus on the relationship between logical topology and quality of service on multimedia flows. To achieve this objective, it will be necessary to modify the current published solutions and to propose new schemes and topologies designed specifically for this type of traffic.

- **Design a network architecture to be used as a reference model for the development of management multimedia traffic protocols and applications over ad hoc wireless networks.**

  Links using cable technologies like Gigabit Ethernet on local networks or Dense Wavelength Division Multiplexing (DWDM) fiber optic links in long distance connections guarantee multimedia traffic by applying conventional techniques of quality of service. They focus on classification and marking packets using some fields on the link layer and the IP protocol. However, when we are working with wireless technologies, which use shared media, or when we have to manage an infrastructure that combines several technologies in the same physical space, these kinds of techniques can be mandatory but they are not sufficient. To approach our goal, first we need to build a framework that allows us to have the appropriate reference for the further development of protocols and applications. So, they will be able to be applied to different technologies and environments and adapt to new technology advances arising in the future without having to completely redesign the particular protocol or application.

- **Develop a communications protocol that implements the components and functionalities of the proposed architecture.**

  The protocol specifications describe how the architecture can be implemented in real networks. In order to prove the validity of both the architecture and the protocol, a software application must be programmed. This application can be used on wireless devices in different laboratory set-ups where we can perform tests and take measures. To achieve this objective, results should confirm that the protocol improves the quality of service on multimedia transmissions when the measures are compared with other systems or topological arrangements.

- **Optimize the protocol operation to be used on wireless sensor networks.**

  Lately, the sensor wireless ad-hoc networks have received much attention because of the potential use we can make of their unused resources. In our case, we are interested in exploit them for multimedia traffic. Therefore, the ultimate proposed objective inside the scope of this research is how to manage the integration of specific issues of sensor networks and sesnor devices in the architecture and protocol proposed. The final application that we are going to develop provides us many practical uses in the world, both in the field of public research and the industry private.

- **Validate the designed architecture and protocol in a test bed using real conditions.**

  Both the architecture and the protocol need to be validated through experimental tests. Since the main objective is to improve the quality of multimedia delivery in real applications, a suitable experimental strategy should be planned to extrapolate the results to the challenging conditions of existing data networks like the Internet. First, each component of the architecture should be tested separately. Then, the whole architecture, including all elements, will pass through several performance tests. Finally, the proposed architecture be tested in real networks.

1.3 Methodology

The whole of activities performed in the research process in this thesis may be divided into three different phases considering the specific objectives, the material used and the methodology. In this section, the methodology used in each phase will be described.

1.3.1 Gather Information

In the first phase the aim was to gather as much knowledge as possible about how the ad-hoc wireless network works and to understand its relationship to multimedia traffic. To achieve it, two main sources of information were used:

- **Scientific literature**

  First we conducted a deep search of the sources that may be relevant to this research:

  - Research magazines and journals specialized on the subject.

- Recent books on wireless ad hoc networks.
- PhD theses related with interest points of this work.
- Standards of protocols and technologies approved by international regulatory bodies such as IEEE or IETF.
- Information published by manufacturers of telecommunications industry.

In this investigation have been handled concepts and technologies from two different fields into the telecommunications area, on the one hand, wireless devices and ad hoc networks and, on the other hand, the quality of service of multimedia streaming. A great amount of valuable information was found, so a classification process was carried out according the main points at which we were interested:

- Range of available wireless technologies currently. The search was focused on the IEEE 802.11, IEEE 802.15.1, and IEEE 802.15.4 standards.
- How these technologies work into the physical layer of the OSI model. In this point, information about the processes of encoding, frequency bands, medium access techniques and algorithms to solve the problem of collisions were collected.
- Link data layer protocol standards. Headers, messages exchange process and state-finite machine were the focus in this category.
- Tests performed in various scenarios: presence of interference or obstacles, geographical distribution of nodes, frequency bandsused, density and pattern of traffic.
- Wireless ad hoc networks. Information about their characteristics, group creation and types of topologies.
- Wireless Sensor Networks: Performance, applications and limitations of sensor devices.
- Characterization of multimedia traffic transmitted through IP networks, including the processes of sampling, encoding and IP packetization.
- Quality of service and quality of experience. Different techniques for measuring the quality parameters, thresholds parameters for

each kind of application and the most popular models of quality of service used.

- Retransmission of multimedia traffic over ad hoc wireless networks. Proposed algorithms and protocols in the literature to improve the performance and ensure the quality of service.

Most of these listed items were successfully covered with the material found in our research. Nevertheless, there was barely some information related with the last point, which is our focus of attention and the objective of this thesis. Thus, the collected information was necessary to contextualize the problem addressed, although it was evident that it would be mandatory complementing it by other means.

- **Experimental tests**

The experiments performed in a laboratory set-up with controlled devices and real traffic allows us to validate the data collected in the previous point and, moreover, complement this information with the measurements results obtained in the laboratory environment.

Protocols and third party applications available under GNU license are chosen when they are available. A model of network using real devices in different scenarios and configurations was constructed to carry out the measures. Among others, the next transmission tests were performed:

- Concurrent telephone calls using a PBX to implement IP telephony with different compression codec.
- Videoconference conversations point to point and point to multipoint.
- IPTV, broadcasting video signal using multicast.
- Video on Demand (VoD), video and audio traffic sent to multiple users via unicast independent connections from a single server.

The results obtained in each of the scenarios can be divided in two types of measures:

- **Objectives.** Measures of quality of service that were obtained using software applications such as network analyzers and sniffers in order to estimate the next parameters: latency, jitter and packet loss.

- **Subjectives.** Quality perceived by the end user when he receive audio or video signal compared to the original signal. The MOS scale was used.

1.3.2 Architecture design

The second phase focuses on the analysis of collected information to design of an architecture that helps us to get the main objective research. The components of the architecture must be able to handle two kinds of data from the network:

- **Wireless devices Information**: Geolocation, CPU and RAM, resources available bandwidth and specific characteristics of the wireless technology used, such as the number of channels and frequency band used.
- **Multimedia traffic information**: quality of service requirements (latency, jitter and packet loss), priority, source and destination of the multimedia transmission.

Different types of components can be distinguished in the architecture. They can be hardware or software and they are added to the model based on the objectives and the data collected:

- **Specific roles**. Initially, all devices in the ad hoc networks have the same function. Nevertheless, the architecture can assign certain roles to specific nodes inside the network. The role allocation is a dynamic process and any node can take on any role at any given time.
- **System processes**. Describe the actions that each node can perform in the network according to the assigned role. For example: The procedure for neighbour discovery, make or break neighbourhoods with adjacent nodes, assign resources to multimedia traffic flows and retransmit audio and video streams.
- **Decision algorithms**. They represent the network intelligence. The system uses them to know what processes must run at all times. Different algorithms are used to determine when and with whom a neighbourhood should be established, to know what are the roles assigned to a node, to

compute the optimal path for a specific broadcast or what amount of resources a transmission needs.

- **Statefull tables**. They gather information from devices and network traffic, then, it is processed and shared with other neighbour nodes. Data tables are updated in a regular basis. The algorithms use the information located on local tables for decision making, so, information on all nodes must be consistent, i.e., the convergence of the network should occur so that the information can be considered valid in the whole network.

The practical implementation of the architecture by designing a communication protocol will be performed in the final phase of the research, however, in this second phase, testing tools are required to validate the proposed model and make the relevant settings. For this purpose, mainly two techniques have been used:

- **Simulations.** The simulation of the entire system is performed running a software application on a single computer. This simulation includes the transmission of multimedia traffic through a number of virtual nodes with the same features and operation which would have the equivalent real devices. This is an easy way to observe the behaviour of the different components of the architecture and identify its shortcomings and failures. The best advantage of using simulations techniques is we can make changes and improve the results really quickly. As disadvantage we must consider that the traffic measures taken on simulated scenarios not always agree exactly the equivalent measures in real scenarios. These small differences can be highly significant when the quality of service parameters must be determined and controlled.
- **Partial implementations**. They are small pieces of programming code. Their function is to make a deeply examination of the behaviour of some architecture component, but without incorporating the rest of components. For example, this kind of programme was made when we are only interested in studying processes such as neighbour discovery, regardless of the other processes in the node. The advantage of this type of solution is it is running on real devices allowing us to detect problems that would go unnoticed in a simulation. The evident disadvantage is that we can not see the whole architecture running.

1.3.3 Communication Protocol Development

In the third and final phase of this research the practical implementation of the architecture is performed. This stage includes two complementary tasks:

- Design of the protocol.
- Application programming.

The communication protocol is the common language that different system devices use to exchange information. Its design requires defining the following items

- **Header protocol**. The information fields managed for each packet are established in this point. Header definition includes the format used for the data (ASCII or binary), size of each field and possible values.
- **States**. All possible states in which a system component can be found are listed here. This information is completed with state diagrams that show the allowed transitions between different states. These diagrams are also known as finite state machines.
- **Messages**. The primary function of a protocol is to create a language that allows the network nodes to communicate between them. It is a key point to define consistently the right way to interpret the information sent and received in the header fields. The list with the different types of available messages determines what and what not the nodes can say.
- **Message exchange**. When two or more nodes are communicating, the process requires the coordination of all of them by sending messages. Using message exchange diagrams indicate the types of messages, message content and the order in which they are sent from each node to complete each process.

After the protocol design is completed, the programming language to make the software implementation must be chosen. Because of their advantages the JAVA language has been selected:

- The compiled code runs on the Java Virtual Machine (JVM) and then it is independent of the platform and operating system used.

- Support for Object-oriented programming (OOP) that facilitates the reuse of parts of the code and simplifies the process of debugging and code optimization.
- Support for network programming at the application level providing several libraries that enable a very fast implementation of network applications.
- Extensive documentation, well organized and available online.
- There are several powerful free software platforms available for programming. In the scope of this research, Eclipse has been used as development environment.

1.4 Dissertation Structure

The dissertation is structured as follows.

Chapter 2 reviews the basic knowledge and concepts that are related to the topics included in the dissertation. It provides the reader a background to understand the matters researched in the PhD. Moreover, it includes the state of the art in the five main topics researched during this PhD, which are multimedia delivery in heterogeneous networks, QoS in wireless ad hoc networks, multimedia and video delivery in ad hoc networks, fault tolerant mechanisms in wireless ad hoc networks and clustering algorithms for wireless ad hoc and sensor networks.

Chapter 3 includes our designs of network architectures and algorithms for multimedia delivery in heterogeneous networks. It has been our starting point of our research and includes three main parts: (1) development and implementation of an architecture to connect nodes in multimedia networks, (2) an architecture and protocol for content delivery, which is based on a group structure, and (3) a cluster-based structure for wireless sensor networks. All of them very related with the aim of the research of this Thesis.

Chapter 4 shows our architecture and protocol design for Multimedia ad hoc networks. In the first part we design a planar architecture, which is extended in the second part with a cluster-based architecture, named MWAHCA. The chapter includes the analytical model, the designed algorithms and the protocol operation.

Chapter 5 presents an extension of MWAHCA, which includes fault tolerance. The fault tolerant mechanism, the protocol extensions, and the analytical model are described in detail.

Chapter 6 describes how MWAHCA has been optimized for its operation in WSNs. We have designed the system architecture, the protocol fields, data structure, message table and the system operation for this environment.

Chapter 7 includes all the performance tests made to validate the design and models included in chapters 4, 5 and 6. These experimental results are taken from the test bench where these deployments were running.

Finally, Chapter 8 draws our conclusion and future research. It also includes the list of publications derived from the PhD.

Chapter 2. Foundations and State of the Art

2.1 Introduction

The aim of this research work has been the development of a new network architecture and a network protocol for multimedia transmissions over wireless ad hoc networks. In the next section of this chapter we are going to introduce the main concepts and definitions into the scope of this objective. This work is related with some research fields like wireless network technologies, multimedia transmission or quality of service models. There are lots of sources on scientific literature where these topics are deeply explained. In consequence, only those concepts, characteristics and functions needed to best understand the rest of the contents in this dissertation will be exposed here.

In the second part of this chapter we focus on the state of the art of latest research on multimedia transmission over wireless ad hoc networks with the aim of giving the appropriate context to our proposals. Because of the proposal has been applied to both ad hoc wireless networks and sensor wireless networks, we will cite the related works in both type of networks.

2.2 Foundations

2.2.1 Wireless ad hoc network

There are many available technologies to build a network infrastructure, but in the scope of this research, we are only interested in wireless networks because they are widely extended and they involve own interesting challenges when multimedia traffic is involved. There are many scenarios where wireless technology overcomes the wired network limitations of connectivity, but at the same time it needs to face other constraints like power consumption restrictions, physical media sharing or resource scarcity. Although some results that we are obtained in this research can be easily extrapolated to wired networks, the models and protocols proposed have been developed specifically for wireless networks where, currently, there is a lack of successful mechanism to guarantee multimedia transmissions. Therefore, our scientific contribution wants to give an answer at the recent demand for new multimedia services that are looking to use the new wireless networks and devices, like smartphones or sensor devices.

The network architecture proposed in this research has been developed by collecting issues and characteristics from the main wireless technologies with the aim of we can use it over a wide range of different wireless technologies. The considered wireless technologies have been:

- **IEEE 802.11**. Currently, it is the most extended wireless technology, most computers and mobile devices have incorporated an 802.11 network interface. By one hand, the standardization from IEEE organization of the Wi-Fi alliance proposal has allowed hundreds of manufacturers to build compatible devices. By other hand, the standard has been improved with new features on quality of service, security and speed transmission.

- **IEEE 802.15.1**. It is a WPAN (Wireless Personal Area Network) technology designed to the connection of devices in a small geographical area, usually least of 10 meters. The greater disadvantage is the bandwidth constraints; although in the last version, 4.0, the transmission speed can reach 24 Mbps. Bluetooth becomes an interesting solution for ad hoc wireless network because of its low power consumption and cheap prices.

- **IEEE 802.15.4**. It is based on the IEEE 802.15.4 standard. The main advantage of this technology is its really low power consumptions; it has become in the best solution when wireless devices use batteries and they are hard to replace, for example because of their location. Also, it is a scalable solution where it can build networks up to 65.535 nodes distributed in segments of 255 nodes. The available bandwidth is really low; in consequence, these devices only can be used for voice transmission or for video with low quality and resolution, like webcams.

For our purposes, the wireless technologies can be classified following the way as nodes into the network are organized:

- **Ad hoc**. The whole network is set by using only wireless devices which are distributed over a geographical area. Each wireless device into the network can communicate directly with any other device, as long as they are into its radio transmission scope. There is not any central device in charge of the management of the network operation. The network can be easily expanded or reduced by adding o eliminating some wireless devices.

- **Centralized**. In this kind of network configuration there is at least one device with special functionalities that are needed to manage the whole network. Thus, two types of devices can be found: Control devices, used to manage the network, and end devices, that communicate between them through the control devices. End wireless devices cannot communicate directly between them in so far as the intervention of the control device is mandatory. Size and scope of the network will be limited by control device capabilities; when new devices are added at the network topology could be necessary to increase the number or the capabilities of control devices. Both control and end devices use the same wireless technologies to build a single network segment.
- **Infrastructure.** This type of wireless network uses hybrid technologies. Several wireless network segments that use a specific technology can be joined by devices from other different technology, even a wired technology. This class of networks provides the most scalable solution because of wireless network fragmentation allows the whole network to grow fast and easily, but the management becomes more complex due to the fact that two technologies must be configured.

Only the ad hoc wireless network will be considered in the rest of this dissertation. Proposed solutions in this thesis can be applied on different wireless technology implementing ad hoc structures, but all devices in a particular ad hoc network need to use the same wireless technology. A given wireless ad hoc network can be described by some characteristics and functionalities:

- **Physical topology**. Geographical distribution of the wireless devices along the three-dimensional space. Distance and transmission power establish the number of nodes of the same network segment into the scope of a device. Into a wireless ad hoc network, a device can communicate directly with other nodes into the scope and indirectly (through other devices in the same ad hoc network) with the other devices beyond the scope. In the last case, a routing algorithm has to be applied on the whole network to keep the network continuity.

- **Logical topology**. Wireless devices into the same geographical space are sharing the same physical media but they can select their internal logical organization. A device can choose to communicate only with a portion of nodes into its scope to build a logical infrastructure independently of the physical structure. The logical relationship between two neighbor devices is called adjacency. The typical organization for wireless ad hoc network is based on groups; the whole network is divided in smaller pieces called clusters, each of them is built by some kind of adjacency aggregation. Simultaneously, different clusters can communicate between them building a hierarchical structure more easy to manage and that will be able to grow fast.
- **Distribution pattern**. Devices can be distributed around the space following several criteria. In experimental conditions two kind of distribution are mainly used: Uniform and random. In real environment the criteria could be determined in function of the particular application used or the geographical constraints. Protocols and algorithms developed to work over ad hoc networks must be able to adapt to these particular patterns.
- **Node density**. It is defined as the average number of devices by area unit. When an ad hoc network has a higher node density it means there are more devices into the scope of each of them and then more options to build logical adjacencies and more flexibility to build clusters. By other hand, a higher node density may cause a higher level of interferences and collisions.
- **Mobility**. It depends on the nature of the devices and the particular use of the network, nodes can have a static location or they can move on the space along time. Mobility ad hoc networks, like vehicular ad hoc networks (VANETS), presents more complex challenges as long as they require more dynamic algorithms to manage the logical topology. In this kind of networks, adjacencies are being continuously created and eliminated. The number, size and structure of clusters also will be modified dinamically.
- **Power consumption**. Many ad hoc networks are built with isolated wireless devices working with batteries. In so far as the replacement of the batteries can be complicated and expensive, the power consumption becomes a key factor when we are designing algorithms and protocols for

these networks. Manufacturer and model specifications show the range interval and main working references, but the logical topology and the algorithms designed will also be really important at this point.

- **Radio covertures**. The device scope will be mainly determined by the transmission power, but other factors should also be considered: Presence of obstacles, level of radio interferences and the kind of antennas used: Omnidirectional, sectorial or directionals.

Finally, there is a particular kind of wireless ad hoc network which we will pay special attention; they are the wireless sensor networks. They are built exclusively by sensor devices with a wireless interface or sensors connected to a wireless device. In this case, there are several additional constraints that must be taken into account: The physical topology is designed for other purpose different from multimedia communications and, therefore, some bandwidth resources are spent for its main purpose of sending sensors data.

2.2.2 Quality of Service (QoS)

When multimedia traffic is sent over switched networks, like current IP data networks, specific requirements of real time transmissions and bandwidth consumption must be considered and controlled. Data packets share the same network infrastructure and physical resources that fragments of video or voice encapsulated with similar protocol headers. To achieve our objective of multimedia transmission over ad hoc networks, the Quality of Service (QoS) parameters must be measured to fit previously established thresholds for each multimedia service guarantee.

For example, an audio or video real time communication between two or more users, like a telephone call, through the network needs to keep the delay time from end to end below 300 milliseconds to achieve an acceptable quality level as perceived by the end users. By other hand, a user downloading a video file for the Internet requires a big amount of bandwidth, but each individual packet can get a big delay time, even they can get to the destination in different order or some packets can be lost and then recovered. In these cases, QoS techniques should be introduced into the network to guarantee the appropriate service level for each kind of traffic. In general, when voice, video and data join in the same output queuing system of an output interface with limited resources then it becomes a great challenge to provide the best QoS of each traffic flow. The difficulty is even greater when we consider wireless ad hoc networks as transport infrastructure. There are several well known QoS models that can be applied on IP networks implemented with wired technologies, but they are not able to guarantee multimedia traffic over wireless environment. That is why we have focused in this research on designing the network architecture to develop the quality of services mechanisms that allow us to solve this problem.

In order to implement any kind of quality service improvement for multimedia traffic, first we must know the characteristics of different multimedia service and their resources requirements. Some issues should be taken into account related at multimedia traffic:

- **Audio, video or both**. An example of single audio communication is a VoIP call transmitted over an IP network, like the Internet. Other multimedia applications use only a video streaming, like video vigilance camera systems, where several video signals are sent from different devices to a central server. In many other multimedia applications and services, audio and video streaming are simultaneously managed. In these last cases we have to identify if each media signal is transmitted on independent flows or instead they are mixed in a single flow.
- **Unidirectional or bidirectional**. A unidirectional communication comes from a source node in the network, which can be identified as multimedia server, and it goes to the end device where is reproduced; the end side will be called multimedia client. Internet users watching online video on the Internet shows this kind of services. In a bidirectional communication, both end devices run simultaneously the server and client roles. It is a peer-to-peer communication usually between two or more users on a video or audio conversation.
- **Point to point or point to multipoint**. In function of the destination, multimedia traffic can be classified as point to point, when a source packet is sent to a single destination, or point-to-multipoint, when a single source packet is copied on the network devices and it is simultaneously transmitted to several destinations in different locations.
- **Codec, resolution and compression rate**. These parameters together will determine the bandwidth resources requirements to keep the right QoS level. Changing resolution o codecs features can dramatically reduce or increase the bandwidth spent by a particular multimedia flow. They can

also have great impact over other QoS parameters like delay or jitter. They are established at the source side.

- **Real time requirements**. Usually, bidirectional communications require to be considered as a real time communication because people can be found at both communication ends keeping an online conversation. Meanwhile, unidirectional multimedia streaming does not need be treated as real time because there is not strict delay constraints.

There are four main network parameters that should always be managed in a QoS politics to achieve their objectives:

- **Bandwidth**. Can be defined as the channel capacity to transmit binary data. The unit of measure for bandwidth is bits per second (bps). Most current technologies provide enough bandwidth for a single multimedia communication but it is mandatory to manage the way this valuable resource is distributed among all the flows. Even in the fast current wireless connections, like 802.11n which can reach 300 Mbps, there will be some sporadically queuing congestion due to the statistical way the packet get to the devices. These small congestions have not effect over common data packets but they can be hugely negative for the QoS of multimedia communications. There are several concepts related at this point:
    - **Bottleneck**. It is called the point in the network where more likely congestion happens. Two conditions can produce a bottleneck, first, when traffic from several input interfaces are followed over a same output interface, and second, when traffic from a fast input interface is sent over a slower output interface.
    - **Path bandwidth**. When packet have to cross several network segments, each of them when different bandwidth, the path bandwidth is defined as the minimum bandwidth on the path. For example, if there are five hops between the source and the destination, four of them are of 56 Mbps and the fifth is only 11 Mbps, then the path bandwidth will be 11 Mbps.
    - **Shared bandwidth**. In a switched network the available bandwidth is shared between all flows, thus the bandwidth by flow can be estimated by equation 1.1.

$$Shared\ Bandwitth = \frac{Path\ Bandwidth}{Number\ of\ Flows} \qquad (1.1)$$

- **End to end delay**. It could be defined as the time interval needed for a packet to travel from the source to the destination. The unit of measure is the second, and it is often expressed in milliseconds (ms). Since the delay time introduced by every device into the path is added to calculate the final end to end delay, the number of hops will be the most relevant factor for the values of this parameter. Each process and retransmissions in each device increase the final delay, the most important are:
  - **Processing delay**. It is introduced by each intermediate device in the path by the time interval between that a packet is received and it is followed to the output interface. Values of this parameter depend on hardware characteristics, CPU speed and processes run as routing or filtering.
  - **Queue delay.** When the packet is followed to the output interface it must wait in a queuing system. The average wait time depends on the kind of queuing system selected, how it is configured, but also the number of flows into the same queue, the queue size and the average packets on the queue. Exact values for a single packet cannot be previously calculated, but average values can be estimated by statistical techniques.
  - **Serialization delay.** Packets are put in the physical media bit a bit following the serialization process. The delay introduced in this stage depends on the size of single packets and the transmission bandwidth of the output interface. For example, when an 1.375 bytes size packet is sent on a 11 Mpbs wireless interface will introduce a delay of exactly 1.375 x 8 / 11.000.000 = 1 ms.
  - **Propagation delay**. It is given by the interval time spent for a single bit to travel from a device to the next one in the path through the physical media. In wireless communication the air is the media and the propagation speed is the light speed on the air, roughly $3x10^8$ m/s. Propagation delay value only depends on distance between source and destination devices.

- **Jitter.** This parameter is defined as the delay variation between two consecutive packets into a single flow. Like delay, it is measured in milliseconds (ms). Jitter is mainly produced by statistical differences in queue occupancy from packet to packet. A packet can find a queue empty but the next packet finds there are some packets from other flows waiting to be transmitted before it do. Jitter may affect really negatively to QoS results even jitter values used to be significantly smaller than delay values.
- **Lost packets**. Packets can be damaged on transmissions over shared physical media, like air in wireless technologies, and they will be dropped in the next hop in the path. Packet can damage when two or more wireless devices start sending in the same channel or when there are other external radio interference sources. Other cause of lost packets happens when free space into a system queue runs out; the new arrived packets have to be dropped. By other hand, when we are considering multimedia communications, packets arriving out of order or with an excessive accumulate delay are also considered lost packets for many services.

2.2.3 Quality of Experience (QoE)

Finally, we have to distinguish between the multimedia signal quality measured on the network with Quality of Service (QoS) parameters described above with the quality perception of end users on the destination side when the multimedia data is reproduced; this is called Quality of Experience (QoE). Because QoE depends on user perception it is inherently subjective quality estimation, unlike QoS objective measures.

Usually, QoE is measured by users sharing the multimedia streaming received on the destination with the original multimedia source. User perception depends on factors such as:

- Distance
- Video size and resolution
- Deep color
- Brightness

- Contrast
- Color saturation
- Distortion
- Naturalness of pictures
- Definition
- Pixel errors
- Partial or full lost frames

We can classify the QoE measurement process following the next:

- **Subjective**. People under experimental conditions evaluate the quality of multimedia audio or video streaming. It is the most precise measure of QoE due to the fact it matches the user perception of real people. There are some standards scales that help to collect measures and share them with results of other research works. The most used is the MOS scale where users fill a survey giving an assessment based on a five point scale:
  - Excellent (5)
  - Good (4)
  - Acceptable (3)
  - Poor (2)
  - Bad (1)
- **Objective**. Electronic devices are used to measure some issues directly related with quality video or audio communications. They use a set of techniques based on human perception pattern and video signal parameters. In general, Objective methods compare video received with video transmitted and extract information to compare with previous database obtained from human answers. In this category we can find the *Perceptual Evaluation of Speech Quality (PESQ)* method. The interval range for PESQ values goes from 0,5 to 4,5. In this range, the PESQ values are very similar at MOS measures.
- **Estimated**. QoE values are estimated from QoS parameters. An example is the *Transmission Rating, R,* based on the ITU G.107 standard, called E model. It is calculated by using fifteen parameters including network

parameters like delay, jitter, lost packets and echo, but also codec information

2.2.4 Multimedia Traffic

Nowadays, there are hundreds of services and applications using multimedia data in different ways and combinations. The main objective of this research work has been to develop an architecture that would be able to adapt the logical network topology and device configurations to optimize transmissions of any type of multimedia flows, i. e., we are not interested in a particular multimedia service but the set of possible services that can be found together on the same environment. Thus, the first stage in this process is to identify the most extended multimedia services with the aim of studying its main characteristics.

First, we have to understand the conversion process through which an analog audio or video signal becomes a stream of network packets.

- **Sampling.** Multimedia samples are periodically taken in order to digitize an analog signal. This process is the most relevant factor in the final quality of the user experience because it establishes the maximum precision the signal on the destination side can achieve. For example, following the Nyquist theorem, on telephone calls the most often sampling frequency used is 8.000 samples by second, in aim to keep the classic 4 KHz telephonic quality sound on the destination. In video transmissions the sampling frequency is established by the Frames Per Second (FPS) parameter, for example a value of 25 FPS is used in PAL video format where 25 pictures are taken by second.
- **Quantizing**. The next step is to assign a binary value at each sample. Finding the appropriate size of the binary number for each sample becomes an interesting problem; if higher size is used higher level of quality but more bandwidth consumption there will get. A typical value of 8 bits per sample is used in audio signals, for example as PCM codec uses. In video signal the quantizing process involve two different parameters: Image resolution and color deep. Each video sample has a horizontal and vertical size, measured and pixels, and the RGB color of each pixel is quantified with a binary value, for example one byte. Thus, we need three bytes for pixel in a RGB picture.

- **Encoding**. With the purpose of maximize the quality of the audio or video signal the encoding algorithm decide the way the quantifying process is carried out. For example in audio PCM encoding a logarithmic function is used to give more precision at low volume signals.
- **Compression.** Most current codec use compression algorithms to reduce the size of digitized signal and save network bandwidth. Because real time nature of multimedia communications these algorithms need to be as fast as possible, in consequence, they do not keep the initial quality of the signal. In general, the more a signal is compressed less quality will be obtained on the destination. This rule is valid for both, audio and video multimedia data.
- **Packetization**. The last stage in this conversion process is to cut the continuous stream of binary data in small fragments to meet the size packet restrictions. Each packet sent to the network their need to add its own headers, thus the bandwidth spent will grow with the number of packets. By other hand, the end to end delay for a single packet will be increased when the size of the packet rises. On audio communications, a balance between size and delay can be found because both the number of packets and the bandwidth required can be kept under low levels. But, in high quality video streaming the packet size will approach the MTU network interface to achieve better use of available resources.

Multimedia services can be grouped in two main categories: Audio and Video. Many parameters, like codec used or resources requirements, are quite similar in most audio communication but different from video traffic. In general, video flows spend more bandwidth resources and they require a different treatment that audio transmissions. The main audio services are:

- **Voice over IP (VoIP**) calls. These are only audio bidirectional communications. They try to replicate the traditional telephony service embedded into the IP network. Hardware and telephony wiring can be replaced by an IP service without need to keep two different technologies. As happens on traditional telephony, only the 0 to 4 KHz frequency range audio signal matching the human voice is kept and encoded. VoIP is a real time bidirectional service with requirements of latency, jitter on both channels.

- **Piped music**. The frequency band sampled and encoded is wider than in telephony calls because of music quality must remain the more similar to the original as possible. Therefore, more bandwidth amount will be spent. This is a unidirectional communication with bandwidth requirements, but delay and jitter do not need to keep values as values as VoIP because it is no real time transmission.

- **Radio over IP**. Similar to previous service where the server is often located on the Internet. Because there is only a server transmitting the audio signal to many clients, multicast techniques can be applied to save bandwidth on trunk links.

- **Public address system (PA)**. It need lower quality of service levels that radio or pipe music because human voice is often the only audio transmitted. It operates into the private corporative network and can enhance its performance with multicast techniques. PA is an unidirectional and no real time service.

By other hand, there are several multimedia video services we must cite:

- **Video over Demand (VoD).** This service is both audio and video streaming going from a video server to a single client. Each client can connect independently to the server and download the video it wishes. It is a no real time application because the video can be buffered before to be reproduced on the destination side. VoD is a unidirectional service with high bandwidth requirements proportional to the number of connected clients.

- **IP Television (IPTV).** The main difference with the previous service is that all clients receive the same video transmission; therefore multicast techniques can be implemented.

- **Video conference**. This is a real time bidirectional communication service. Similar at call telephones but they incorporate video signal. High bandwidth requirements join the delay and the jitter constraints.

- **Videogames.** Online videogames have shown up as a new multimedia service on IP networks. Players share a same virtual world where they can interact, talk and fight between them. In this case, data packets carrying the virtual world information must also be treated as real time traffic, like audio and video.

Undoubtedly, one of most challenging characteristics for multimedia traffic is the real time requirement. As opposed to common data packets, like file transfer or web browsing, multimedia packets must be delivered to the destination following stricter criteria:

- **Delivery in order**. In real time applications packets must be processed immediately after they are received on the destination side. Thus, when the packet number N in a sequence is received, then the N+1 packet is expected but, if the N+2 arrives before then it will be processed and the N+1 skipped even it gets to the destination later. In consequence, any out of order packet could be considered as a lost packet.
- **Minimum delay**. Real time constraints establish delay thresholds specific for each multimedia service. Packets over this threshold will be dropped.
- **Minimum delay variation.** With real time applications packets must be continuously delivered to the destination. Even some delay is allowed in these applications under the threshold limit, the flow cannot be interrupted and jitter, if it exits, must be eliminated. A jitterbuffer is usually set on the destination end to suppress its negative effect.
- **Minimum lost packets**. In real time applications, recovery mechanisms for lost packets are not implemented because final delay will possibly increase above the delay threshold. However, some percentage of lost packets can be allowed. Some codec solve the lost packet problem implementing redundancy techniques; a lost packet can be recovered on the destination side by the redundant information in previous and next packet.

The scope of our research the IP network, like the public network of Internet or the corporative private network implemented in most companies, have been include. However, this dissertation is not limited to the TCP/IP protocol stack. We are focus on wireless ad hoc network and the associate technologies, like IEEE 802.11, IEEE 802.15.1 or IEEE 802.15.4, therefore other network and application protocol are also considered.

With the purpose of providing the best service for multimedia transmission on IP networks two protocols are preferentially selected because of their functionalities.

- **UDP**. It is the transport protocol usually selected for real time multimedia traffic. This protocol does not offer any mechanism of flow control or reliability. There are not number of sequence or acknowledge capabilities included on the protocol header. It is preferred because of delay behavior of UDP packet meets the real time requirements.
- **RTP**. The TCP protocol is discharged as transport protocol because its reliability and flow control mechanism are not appropriate for multimedia services. However, some functionality, like providing numbers of sequence, are required to meet some of the real time requirements, like packets delivery in order. The Real Time Protocol (RTP) is used for most applications to complement the UDP protocol and provide additional multimedia information to the destination side about the carried packet

2.3 State of the Art

In last years, research on multimedia distribution over wireless ad hoc networks has been increased and a big amount of applications have begun to focus on them. On one hand, new multimedia services have emerged and they require more and more network resources. On the other hand, networks have become more complex because currently we can find a wide range of different devices and wireless network technologies offering new capabilities and resources.

The scope of the dissertation has brought us to split de state of the art in several parts. The first one is related with multimedia delivery in hererogeneous networks. The second presents published about QoS in wireless ad hoc networks. The third is focused on preseting last research works about multimedia and video delivery in ad hoc networks, Forth one includes the state of the art of fault tolerant mechanisms for wireless ad hoc networks. Finally, the fifth one is focused on clustering algorithms for WSN. These 5 subsections cover the state of the art of the proposed architecture and protocol and the performed improvements.

2.3.1 Multimedia delivery in hererogeneous networks

There many work that study the interconnection system between surrogates. Some of them are based on P2P systems [9] [10]. There are also GRID-based proposals such as the one presented in reference [11]. Giancarlo Fortino et al. proposed in [12] three models to develop Content Delivery Networks (CDNs) (P2P-based, GRID-based and agent-based systems). The fact of developing a new model means having a new interconnection system among the CDN surrogates. Z. Xiang et al. presented a Peer-to-Peer Based Multimedia Distribution Service in [13]. They propose a topology-aware overlay in which nearby hosts or peers self-organize into application groups. End hosts with in 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. Some studies demonstrate that the overlay network with a flat topology does not scale well [14]. Hierarchical overlay network topology is required for a large-scale CDN to perform content delivery scalable and efficiently. Several schemes can be used to organize the surrogates, through manual configuration or through a self-organizing scheme [15]. Cluster systems are one of the most used schemes when scalability is needed [16][17]. There are some works in the literature that show the benefits of cluster-based schemes for multimedia delivery [18].

2.3.2 QoS in wireless ad hoc networks

Traditional QoS techniques cannot provide us the quality level that multimedia users expect on sending real time multimedia traffic over wireless ad hoc networks. QoS provisioning in ad hoc network is not dedicated to any specific layer rahter it require coordinate efforts from all layers [19] QoS support components includes: QoS models, QoS resource reservation, QoS Routing and QoS Medium Access Control (MAC)[20]. All QoS components must cooperate together in order to achieve the appropirate performance, such as Flexible QoS Model for MANET (FQMM)[21] does; it is based both on IntServ and DiffServ QoS models. Several routing algorithms for wireless ad hoc network are focuse on QoS performance such as Lajos Hanzo et al collect in [22] Since the main weight of maintaining a fast multimedia delivery and an optimum path for the streams in an ad hoc network is carried out by the routing protocols [23], most authors have studied the routing protocols in ad hoc networks in order to know their features and which ones are the most appropriate to provide QoS [24].

There are few works that show real QoS measurements in MANET. We can find some works about multimedia content transmission simulations that analyze the required QoS parameters in these cases. Others propose new architectures and algorithms for transmitting multimedia. In this section we are going to show the main ones.

B. Zhang et al. [25] discuss some key design considerations in QoS routing and present a review of previous work addressing the issue of route selection subject to QoS constraints. They present a new on demand delay-constrained unicast routing protocol. Authors employed various strategies to reduce the communication overhead in acquiring cost-effective delay-constrained routes. Simulation results verify the high performance of proposed protocol.

H. Luo et al. [26] propose the Unified Cellular and Ad-Hoc Network (UCAN) architecture for enhancing the cell throughput while maintaining fairness. In UCAN, a mobile client has both 3G cellular link and IEEE 802.11-based peer-to-peer links. The 3G base station forwards packets for destination clients with poor channel quality to proxy clients with better channel quality. The proxy clients use an ad-hoc network composed of other mobile clients and IEEE 802.11 wireless links to forward the packets to the appropriate destinations, thereby improving cell throughput. They redefine the 3G base station scheduling algorithm so that the throughput gains of active clients are distributed proportional to their average channel rate.

B. Li et al. proposed in [27] an algorithm called NonStop. It is a collection of middleware-based run-time algorithms that ensures the continuous availability of multimedia streaming services, while minimizing the overhead involved. The network-wide continuous streaming coverage is achieved by partition prediction and service replication on the streaming sources and assisted by distributed selection of streaming sources on regular mobile nodes and users. The proposed algorithms are validated by extensive results from performance evaluations.

In [28], P. Sinha presents solutions and approaches for supporting QoS in ad-hoc networks at the physical, Medium Access Control (MAC), and routing layers. He also presents approaches at other layers and describes future challenges that need to be addressed to design a QoS enabled ad-hoc network.

S. Chakrabart et al. provide a brief introduction for guaranteeing QoS in ad hoc mobile wireless networks [29]. Their work is focused on QoS routing. This is a complex and difficult issue because of the dynamic nature of the network topology and generally imprecise network state information. They present the basic concepts and discuss some results.

P. Mohapatra et al. [30] provide a survey of QoS support in MANETs. In addition to the basic issues in QoS, they describe the efforts on QoS support at each communication layer, starting from the physical and going up to the application layer. Few proposals on interlayer approaches for QoS provisioning are also addressed.

A. M. Abbas et al. [31] discuss methods for QoS provisioning at different levels including routing, MAC and cross layer. They also discuss schemes for admission control and scheduling that are proposed in the literature for QoS provision. They compare salient features of various solutions and approaches and point out directions for future work.

P-H. Chuang et al [32] propose an approach to support QoS and multimedia applications in ad hoc wireless network. But they only presented a proposal with some 1 hop simulations, not a deployment test and simulation.

2.3.3 Multimedia and video delivery in ad hoc networks

In recent years, the research on multimedia distribution over ad hoc networks has been increased. It has happened because of the improvement of the hardware capabilities and the appearance of new multimedia services. In order to support real time multimedia applications, we should mainly take into account QoS constraints.

In [33], R. Zhang et al. propose several improvements over MAC protocols to solve some of the main problems as the stringent Quality of Service (QoS) requirements of video traffic, the limited wireless channel bandwidth and the broadcast nature of wireless medium in ad hoc networks. Authors proposed two conflict avoidance strategies for reservation and contention interleaved wireless systems. With a dual-buffer, the video packets being transmitted using contention or reservation-based channel access are separated and stored in two buffers. Authors also developed analytical models considering the interactions of reservation and contention periods. The performance tests were focused on the contention-based access. Simulation results showed that the backoff strategy can achieve higher throughput when the number of reserved periods in each superframe is large. Authors also checked the results of their proposal when transferring MPEG-4 video streams. The proposed hybrid approach with the two buffering architectures provided a considerably better performance, due to the higher reservation utilization and lower contention level.

T. Mehta and Z. Narmawala used a Video Traffic Model to generate video traffic frames in [34]. They observed that Network Coding performs well in lossy wireless ad hoc networks in both multicast and broadcast scenarios. Even in wireless ad hoc networks with low density of nodes network coding performs well using their multi-copy packet transmission scheme. In their work, each sender node encodes the packet using a variant of Network Coding, which is Random Linear Network Coding (RLNC) with Multi Generation Mixing (MGM), with the aim to provide more protection to I (Intra frame of MPEG 4 video traffic) frames in order to minimize the multiplicative loss by incurring slight delay in transmission. Mixing different types of packets increases the Packet Delivery Fraction as well as reduces Packet Drop Rate and Block Delay of multimedia transmission over wireless ad hoc networks.

Since the main weight of maintaining a fast multimedia delivery and an optimum path for the streams in an ad hoc network is carried out by the routing protocols [23], most authors have studied the routing protocols in ad hoc networks in order to know their features and which ones are the most appropriate to provide QoS [24]. Moreover, some authors have developed QoS-aware routing protocols for ad hoc networks such as the following ones.

R. Al Turki and R. Mehmood in [35] studied video streaming applications over ad hoc networks and analyzed the results obtained through simulations using the OPNET software. They also surveyed the main challenges in ad hoc network research and reviewed the QoS literature for ad hoc networks. They evaluated performance of video streaming applications over ad hoc networks by simulating few scenarios with 5 different routing protocols. The results show that it is possible to support multimedia applications over medium sized networks.

A. Jamali et al. demonstrated in [36] that ad hoc networks can support video streaming. In order to do it, they analyzed some routing protocols through simulations in OPNET environment in terms of multimedia and real time application and QoS. In their study, they analyze AODV, DSR, OLSR, TORA, and

GRP for Multimedia Streaming. Their results demonstrate that ad hoc networks can have good video streaming quality. They conclude the paper stating that designing a multimedia ad hoc network is difficult because of the higher QoS requirement and the kind of network topology.

A. Abdrabou and W. Zhuang presented in [37] a model-based quality-of-service (QoS) routing scheme for IEEE 802.11 ad hoc networks. This proposal is based on a cross-layer design approach. The scheme proposed selects the routes based on a geographical on-demand ad hoc routing protocol and checks the availability of the network resources by using traffic source and link-layer channel modeling. The system also considers the IEEE 802.11 features and the node interactions. The protocol checks if the selected route is able to admit traffic flow without affecting other flows already in service. The simulation results show that the proposal is efficient in resource utilization while satisfying the delay bound probabilistically with a low overhead.

D. Kandris et al. presented in [38] a dual scheme based on the combined use of an energy aware hierarchical routing protocol with an intelligent video packet scheduling algorithm for efficient video communication, which aims at both energy saving and high QoS attainment. PEMuR adopts a routing protocol which is able to the select the most energy efficient routing paths while it manages the network load according to the energy residues of the nodes and prevents useless data transmissions through the proposed use of an energy threshold. In addition, this protocol is able to reduce the video transmission rate with the minimum possible increase of distortion. The simulations performed by authors showed that this proposal prolong the node lifetime. It also enhances the metric of network performance in the case of nodes with non uniform energy distribution while maintaining high levels of the perceived video quality (PSNR).

N. Taing et al [39] proposed a Routing Scheme for Multimedia Services, which selects the shortest path by using power level. This proposed scheme selects a shortest path for multimedia traffic by applying larger power level because the delay is sensitive to such kind of traffic. Moreover, for non-real time traffic, this algorithm uses the shortest path by using power level for non-real time. They conclude that their proposal provides the lower mean number of hops for multimedia traffic than the mean number of hops for non-real time traffic. As the result, the transmission delay of multimedia traffic can be decreased. They also show that its proposal scheme can provide higher throughput for multimedia traffic.

2.3.4 Fault tolerant mechanisms for wireless ad hoc networks

Nowadays, we can find some works on routing protocols to solve the problems we can find in multimedia wireless ad hoc and sensor networks. Most of these proposals are tested in network simulators and very few are developed for real environments.

A. S. Tai et al. [40] developed an algorithm for failure detecting in distributed systems formed over ad hoc wireless networks. By exploiting cluster-based communication architecture, the system is able to make the failure detection service scalable and resilient to link failure. Moreover, the proposed protocol allow intra- and inter-cluster communication to be resilient to message loss and node failure by taking advantage of the inherent message redundancy in ad hoc wireless networks. F. Kuhn et al. [41] also studied distributed approximation algorithms for fault-tolerant clustering in wireless ad hoc and sensor networks. Other authors such as Y. Xue and K. Nahrstedt [42] proposed a new routing service named best-effort fault-tolerant routing (BFTR). The design of BFTR is not to detect whether routing path consists of any misbehaving node, but to evaluate its routing feasibility based on its end-to-end. Analytical and experimental results demonstrate that BFTR greatly improves the ad hoc routing performance in the presence of misbehaving nodes. H. L. Chao et al. [43] proposed a fault-tolerant routing protocol called Sensor On-demand Multi-path Distance Vector Reliable (SOMDV-R) routing protocol for wireless sensor networks which support reliable data delivery. This protocol takes into account reliability demand and link quality to determine the number of desired paths. Results show that, in comparison with Ad hoc On-demand Multi-path Distance Vector (AOMDV) and AODV, SOMDV-R can achieve higher packet delivery rate, lower routing overhead and lower mean latency, upon different channel conditions.

E. Pagani and G. P. Rossi [44] presented a reliable broadcast protocol designed for mobile ad-hoc networks. The proposed protocol was implemented on top of the wireless MAC protocol which sits over the clustering protocol. The protocol also provides tolerance in communication failures and host mobility. The reliable broadcast service ensures that all hosts in the network deliver the same set of messages to the upper layer. Their results show that the proposal provides high broadcast and multicast services as an efficient and reliable alternative to flooding method.

G. Gupta and M. Younis proposed in [45] a high-energy gateway node that acts as a centralized manager to handle the sensors. The gateway serves as a hop to relay data and commands from sensors to a distant node. They have introduced a two phase in the communication process to apply fault-tolerance approach in sensor networks. This system allows detecting and recovering sensors from the failed gateways without shutting down or re-clustering the system. Their approach enables fault tolerance in the system by performing periodic checks on the status of the gateways. Sensors managed by a faulty gateway are recovered by re-associating them to other clusters based on backup information created during the clustering time.

On the other hand some researchers proposed fault tolerant routing protocols for ad hoc networks. Among them, A. Boukerche et al. [46] proposed two routing protocols: periodic, event-driven and query-based protocol (PEQ) and its variation CPEQ. They are two fault-tolerant and low-latency algorithms that meet sensor network requirements for critical conditions supervision in context-aware physical environments. PEQ uses a small amount of information for the routing mechanism (basically the hop level and routing table). When a failure is detected, unlike other solutions that use three way protocols, PEQ broadcasts a SEARCH packet to its neighbors, and receives a reply with their hop level and identification. The neighbor with lower hop level is chosen as the new destination. In this way, loop back is avoided. Other authors as T. Bheemarjuna et al. [47] presented MuSeQoR, a new multi-path routing protocol that tackles the twin issues of reliability (protection against failures of multiple paths) and security, while ensuring minimum data redundancy. Reliability is addressed on the context of both erasure and corruption channels. Authors also quantified the security of their protocol in terms of the number of eavesdropping nodes. The requirements of reliability and security in a session are specified by a user and are related to the parameters of the protocol adaptively. This relationship shows how the protocol attempts to simultaneously achieve reliability and security. In addition, the protocol minimizes the redundancy by using optimal coding schemes and by dispersing the original data.

R. Melamed et al. [48] also presented a simple fault-tolerant protocol called Octopus. This proposal also showed great efficiency in routing tasks for large MANETs which can work with mobile nodes.

Finally, regarding to power consumption, T. Antoniou et al. [49] proposed a new energy efficient and fault tolerant protocol for data propagation in wireless sensor networks. It is called Variable Transmission Range Protocol (VTRP). The main idea of this protocol is based on the varying range of data transmissions. The protocol exhibits high fault-tolerance, by bypassing obstacles or faulty sensors, and increases network lifetime (since critical sensors, i.e. close to the control center, are not overused). The protocol was evaluated and compared, in terms of performance measures and energy consumption, with representative protocols such as Licklider Transmission Protocol (LTP). The results show that their protocol achieves significant improvements in energy efficiency and network lifetime.

2.3.5 Clustering algorithms for Wireless Ad Hoc and Sensor Networks

The overlay ad hoc network with a flat topology does not scale well [14].Thus, nodes into a wireless ad hoc network can be organized in many topology configurations. One of the most used is based on cluster. There are some works in the literature that shows the benefits of cluster-based schemes [18]. Cluster-based networks can be considered a subset of group-based networks where each cluster matches a different group. However, group-based networks can be arranged other different topology that cluster-based. They can be classified according to whether the architectures are based on Cluster Head [50] or on Non Cluster Head [51]. The first architecture needs Cluster Head to control and manage the group, and the second one does not have a specific node to perform this task. According to D. Wei et al. [52] cluster topologies can be classified into four categories: single-hop or multi-hop, stationary or mobile, synchronous or asynchronous, and location-based or non location-based. On the other hand, J. Yu et al. [18] made a categorization of clustering schemes in stationary and mobile ad hoc networks and sensor. They classified 14 proposed clustering schemes into six categories based on their main objectives. Moreover, they discussed each clustering scheme in terms of objective, mechanism, performance, and application scenario, and discussed the similarities and differences between schemes of the same clustering category.

There are two published papers, written by A. A. Abbasi et al. [53] and O. Boyinbode et al. [54], which present a synthesis of existing clustering algorithms in WSNs and highlight the challenges in clustering. They survey different clustering algorithms for WSNs, emphasizing their objectives, features, complexity, etc. They also compare their metrics such as convergence rate, cluster stability, cluster overlapping, location awareness and support for node mobility.

The paper authored by R. Agarwal et al. [55] review several clustering algorithms which help organize mobile ad hoc networks in a hierarchical manner and presented their main characteristics. With this survey we see that a cluster-based MANET has many important issues to examine, such as the cluster structure stability, the control overhead of cluster construction and maintenance, the energy consumption of mobile nodes with different cluster-related status, the traffic load distribution in clusters, and the fairness of serving as cluster heads for a mobile node.

Despite of the aforementioned surveys, we would like to mention 3 clustering algorithms not included in these surveys because of their importance.

L. Ramachandran et al. [56] proposed two new distributed clustering algorithms for wireless ad hoc networks. They presented a 2- stage O(N) randomized algorithm for a N node complete network, which finds the minimum number of star-shaped clusters, all at their maximum size. They also proved the correctness of this algorithm. They then presented a completely deterministic O(N) algorithm in which cluster heads are elected autonomously by the nodes. They compared their performance using simulations on top of Bluetooth's device discovery procedures. Results show that the randomized algorithm performs better with respect to both cluster and network formation times.

M. Chattarjee et al. [57] proposed a weight based distributed clustering algorithm (WCA) which can dynamically adapt itself with the ever changing topology of ad hoc networks. Their approach restricts the number of nodes to be catered by a cluster head so that it does not degrade the MAC functioning.

M. Kavitha et al. [58] proposed an energy enhanced version of the M-SPIN (EEM-SPIN) protocol using WCA for WSNs. It has the flexibility of assigning different weights and takes into account combined metrics to form clusters automatically. Limiting the number of nodes inside a cluster allows restricting the number of nodes catered by a cluster head so it does not degrade the MAC functioning. For a fixed cluster head election scheme, a cluster head with constrained energy may drain its battery quickly due to heavy utilization. In order to spread the energy usage over the network and achieve a better load balancing among cluster heads, re-election of the cluster heads may be a useful strategy.

Next, we review how some of the main cluster-based multimedia ad hoc networks are created.

Y. Huang et al [59] have presented a cluster based model to support multimedia service. The proposed model transmits multimedia streaming stably in ad hoc networks, while mobile users who consume multimedia streams tend towards group-based behavior. An on-demand connection prediction to measure the likelihood of connectivity of cluster-based routes in a future time is applied to the cluster-based transmission of multimedia streaming. They proposed a routing method called PLCBRP (Cluster-based routing with the prediction of connection probability), which combines the cluster-based routing protocol with the prediction scheme. PLCBRP discovers an optimal loosely cluster-based route for transmitting long multimedia streams. Simulation results indicate that PLCBRP delivers more data packets and provides more quality on the transmission of multimedia streaming than other flat on-demand routing protocols do.

S. Tang et al. [60] developed a QoS supporting scheme for dynamic traffic conditions by controlling data generating rates at individual clusters. Besides, they have investigated an explicit solution on the energy distribution at different clusters in the WSN, based on an optimal energy allocation criterion. The obtained network energy distribution formula is particularly convenient for node deployment design in WSNs. The proposed algorithm is presented and validated by numerical simulations. Some situations are also discussed and presented by experimental examples.

D. Rosari et al. proposed MEVI in [61]. It is a smart multi-hop hierarchical routing protocol for efficient video communication over Wireless Multimedia Sensor Networks. It combines a cluster formation scheme with low signaling overhead in order to ensure reliable multi-hop communication between Cluster Heads and Base Stations. For route selection, a cross-layer solution that selects routes based on network conditions and energy issues, and a smart scheme to trigger multimedia transmission according to sensed physical environmental conditions. The cluster approach aims to minimize the energy consumption. MEVI allows the transmission of multimedia content with QoS/QoE support by introducing a hierarchical routing protocol. Simulation experiments show the benefits of MEVI in disseminating video content for large and small field size, compared with Low-Energy Adaptive Clustering Hierarchy (LEACH) and Power Efficient Multimedia Routing (PEMuR) in terms of network lifetime and video quality level.

2.4 Conclusion

In this chapter we have introduced the main concepts and definitions needed to achieve the aim of the Thesis. We have reviewed several research fields like wireless ad hoc network technologies, multimedia delivery, quality of service models and wireless sensor networks. There are lots of sources on scientific literature where these topics are deeply explained. In consequence, only those concepts, characteristics and functions needed to best understand the rest of the contents in this dissertation have been exposed here.

We have also included the state of the art of the latest research on multimedia delivery in heterogeneous networks, QoS in wireless ad hoc networks, multimedia and video delivery in ad hoc networks, fault tolerant mechanisms in wireless ad hoc networks and clustering algorithms for wireless ad hoc and sensor networks.

Chapter 3. Design of Network Architectures and algorithms for multimedia delivery

3.1 Introduction

In this chapter, several models and architectures for multimedia transmission are proposed. They focus on different aspects of network infrastructure or algortihm decision mechanisms to improve the multimedia transmission over several kinds of networks. These works constitute the foundation of the development of the network architecture for multimedia delivery over wireless ad hoc networks, (Multimedia Wireless Ad Hoc Networks Cluster Architecture, named MWAHCA), discussed in the next chapter.

In the second subsection, a new architecture for logic interconnection of different segments and different types of network is presented. It is designed with the aim to create a single continuous topology prepared for distribution of multimedia contents. This architecture can be applied in different environments such as content distribution networks, peer-to-peer or ad hoc wireless networks.

The third subsection proposes a network model to join content distribution networks from different providers. The decision algorithms aim to ensure QoS parameteres in multimedia deliverings. To validate the proposal, it is implemented with real devices,providing experimental measurements.

In the last subsection, the performance of a group-based solution to interconnect content distribution networks is analyzed. The scalability and reliability of the protocol is tested on a controlled laboratory environment, reproducing real cases network conditions. The results are also compared with previous proposals in this research.

In the fourth and last subsection, a cluster-based architecture that allows building a common topology for different wireless sensor networks is proposed.

3.2 An Architecture to Connect Disjoint Multimedia Networks Based on node's Capacity

Today, grouping nodes in multimedia networks is more and more extended in Internet. They could be grouped based on content sharing, or according to similar characteristics, functionalities or even social trends. Many researchers try to find the best application layer protocol and architecture to organize and join nodes in a multimedia network [62] [63]. Once they are grouped, data are kept inside the multimedia network and only nodes joined to the network can download or stream data shared inside. Let's consider a set of autonomous multimedia networks. All nodes in the multimedia networks run their application layer protocol. Each node shares or streams content using the application layer protocol. All application layer protocols can be translated to other protocols and they are not encrypted using signatures or private keys. An interconnection system over those multimedia networks will give some benefits for the whole system such as:

1) Content availability will be increased.
2) It will facilitate data or content replication in other multimedia networks.
3) When a new multimedia network is added to the system, existing desktop applications will not be changed.
4) Network measurements could be taken from any multimedia network.
5) Desktop applications could search and download from every multimedia network using only one open service.

Once the interconnection system is working, when a node looks for some data, first it will try to get it from its network. In case of no result, the search could be sent to other networks. If it is found, the data can be downloaded or transferred to the first network. Once the node has that data, it will act as a cache for its network sharing the data. The existing systems are based on a desktop application which knows several protocols and is able to join several networks. Examples are Shareaza [64], MLDonkey [65], giFT [66], Morpheus [67] and cP2Pc [68]. To use those solutions, taking "search for a file" as an example, the user has to be permanently connected to all networks. The computer that joins all networks needs a lot of processing capacity. In addition, if a new P2P file sharing network is developed, a new plugin is required to support the new architecture; and, all users will have to update their clients to join the new network. On the other hand, this solution allows a node to join several multimedia networks, but those networks will not be interconnected.

3.2.1 Mathematical description

When a node joins the proposed architecture, it has the following parameters previously defined:

1) A multimedia network identifier (networkID). All nodes in the same network have the same networkID.

2) Upstream and downstream bandwidth.

3) Maximum number (Max_num) of supported connections from other nodes.

4) Maximum % of CPU load used for joining the architecture by the desktop application (Max_load).

5) Kind of content or data shared in its network.

Nodes could have 2 types of roles. Nodes with first role (Dnodes) establish connections with Dnodes from other networks as a hub-and-spoke. Dnodes are used to send searches and data transfers between networks. Nodes with second role (Onodes) are used to organize connections between Dnodes from different networks. For scalability reasons, Onodes could have 2 roles: to organize Dnodes in zones (level-1 Onodes) and to have connections with Onodes from other networks (level-2 Onodes). This proposal considers only one type of Onode which has both roles. More details about an interconnection system using two levels of Onodes are shown in [69].

First node, from a new multimedia network, will be Onode and Dnode. New nodes will be Dnodes, but they could acquire higher roles because of some parameters discussed later. A node identifier (nodeID) is assigned sequentially, using a timestamp by the Onode, to every new node. Then, the new node calculates 2 parameters:

- **δ parameter**. It depends on the node bandwidth and its age in the system. It is used to know which node is the best one to have a higher role. A node with higher bandwidth and older is preferred, so it will have higher

δ. The number of nodes needed to promote a node depends on the type of multimedia network. Equation 3.1 gives δ parameter.

$$\delta = (BW_{up} + BW_{down}) \cdot K_1 + (32 - age) \cdot K_2 \quad (3.1)$$

Where *age=log₂(nodeID)*, so age varies from 0 to 32. K_1 and K_2 adjust the weigh of the bandwidth and the node's age. We can see that older nodes will have more architecture functionalities than new ones unless they have low bandwidth. Nodes with high bandwidth and relatively new ones could have higher δ values.

- **λ parameter.** It represents the node capacity, and depends on: the node's upstream and downstream bandwidth (in Kbps), its number of available connections (Available_Con) and its maximum number of connections (Max_Con) and its % of available load. It is used to determine the best node. It is given in equation 3.2.

$$\lambda = \frac{\mathrm{int}\left[\frac{\left(BW_{up} + BW_{down}\right)}{256} + 1\right] \cdot Available\_Con \cdot \left(100 - load\right) + K_3}{Max\_Con} \quad (3.2)$$

Where 0 ≤ Available_Con ≤ Max_Con. Load varies from 0 to 100. A load of 100% indicates the node is overloaded. K_3 give λ values different from 0 in case of a 100% load or Available_Con=0.

Dnodes take the Onode election based on δ parameter. There could be more than one Onode per network. It depends on the number of Dnodes in the multimedia network. From a practical point of view, we can consider a big multimedia network, which is split into smaller zones (different multimedia networks). Every node has 5 parameters (networkID, nodeID, δ, λ, role) that characterize the node. The following describes how every node works depending on its role.

3.2.2 Onodes organization

We have chosen the routing algorithm SPF to route searches for providing Dnodes adjacencies. The cost of the ith-Onode (C_i) is based on the inverse of the i^{th}-Onode λ parameter. It is given by equation 3.3.

$$C = \frac{K_4}{\lambda} \tag{3.3}$$

With $K_4=10^3$, we obtain C values higher than 1 for λ values given in equation 2. Nodes with Available_Con=0 and/or load=100 will give C=32.

The metric, to know which one is the best path to reach a destination, is based on the number of hops to the destination and the cost of the Onodes involved in that path. It is given by equation 3.4.

$$Metric = \sum_{i=1}^{n} C_i \tag{3.4}$$

Where C_i is the i^{th} node virtual-link cost and n is the number of hops to a destination.

Given G = (V, C, E), an Onodes network, where V is a set of Onodes V= {0, 1,…, n}, C is a set of costs (C_i is the cost of the i^{th}-Onode and $C_i \neq 0 \ \forall$ i-Onode) and E is a set of their connections. Let $D=[M_{ij}]$ be the metric matrix, where M_{ij} is the sum of the costs of the Onodes between v_i and v_j. We assume $M_{ii}=0 \ \forall \ v_i$, i.e., the metric is 0 when the source Onode is the destination. On the other hand, v_i and $v_j \in V$, and $M_{ij} = \infty$ if there is not any connection between v_i y v_j. So, supposing n hops between v_i and v_j. The Onode M_{ij} matrix is given by equation 3.5.

$$M_{ij} = \begin{cases} \min\left(\sum_{k=1}^{n} \cos t(v_k)\right) & \text{When } \exists \text{ a path between } v_i \text{ and } v_j \\ \infty & \text{In other case} \end{cases} \tag{3.5}$$

In order to calculate the number of messages sent by the whole network, we consider that the i^{th}-Onode has $CN(i)$ neighbours, where $CN(i)=\{j \mid \{i,j\} \in E\}$. On the other hand, every Onode calculates $M_{ij} \ \forall j$ and builds its topological using Dijkstra algorithm. Then, every i Onode sends its topological database to its $CN(i)$ neighbours and they to their neighbours until the network converges. Finally, all Onodes can calculate their lower metric paths to the rest of the Onodes, i.e., M_{ij} [70]. When there is a fixed topology where all Onodes have to calculate the lower metric path to the others, the number of messages sent by every Onode is $O(V \cdot D+E)$, where D is the network diameter. In the proposed system, Onodes can join and leave the topology any time, so the metric matrix is built as the Onodes join and leave the topology. When a new Onode joins the system, it will build its topological database and its neighbours will send the update to its neighbours except to the new Onode. Neighbour Onodes will send the update to all their neighbours except to the one from who has received the update. In addition, the update is sent only if the update is received from the Onode in the lower metric path to the one who has started the update (Reverse Path Forwarding [71]).

Taking into account considerations aforementioned, given *n=|V|* Onodes in the whole network, m=|E| the number of connections in the network, *d* the diameter of the network and supposing that the i^{th}-Onode has *CN(i)* neighbours, we obtain results shown in table 3.1 (t_p is the average propagation time).

Table 3.1. Results for Onodes organization.

Number of messages sent by an Onode because an update	CN(i)-1
Number of updates sent in the whole network	n
Time to converge	$d \cdot t_p$
Metric	$\sum_{i=1}^{n} \frac{1}{\lambda(i)}$
Number of messages sent by an Onode because an update	CN(i)-1

3.2.3 Dnodes organization

Given *G = (V, λ, E)* a network of nodes, where *V* is a set of Dnodes, *λ* is a set of capacities (*λ(i)* is the i-Dnode capacity and *λ(i)≠0* $\forall$ i-Dnode) and *E* is a set of connections between Dnodes. Let *k* be a finite number of disjoint subsets of *V*, $\forall$ *V=Union(V_k)*. Given a Dnode v_{ki} (Dnode *i* from the *k* subset), there is not any connection between Dnodes from the same subsets (e_{ki-kj}=0 $\forall$ V_k). Every v_{ki} Dnode has a connection with one v_{ri} from other subset *(r≠k)*. Let *D=[M_{ki}]* be the capacity matrix, where M_{ki} is the capacity of the i-Dnode from the *k* subset. Let's suppose *n=|V|* and *k* the number of subsets of *V*, then we obtain equation 3.6.

$$n = \sum_{i=1}^{k} |V_k| \tag{3.6}$$

On the other hand, the number of connections *m=|E|* depends on the number of subsets (*k*), the number of Dnodes in each subset (*km*) and the number connections that a Dnode has with Dnodes from other subsets (we will suppose that a Dnode has *k-1* connections). Equation 3.7 gives *m* value.

$$m = \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k_m}\sum_{l=1}^{k-1} v_{km}(l) \qquad (3.7)$$

Where $v_{km}(l)$ is the l-connection from the m-Dnode from the *k* subset.

Let's suppose there are *n* multimedia networks and only one Onode per multimedia network. When a Dnode sends a request to its Onode, it broadcasts this request to all other Onodes, so, taking into account results from subsection B, there will be n messages. When this message arrives to the Onode from other network, it will elect its two connected Dnodes with higher λ and they acknowledge the message, so, it is needed 2 messages. Then, Dnodes from other networks will send a connection message to the requesting Dnodes and they confirm this connection, so it is needed 2 more messages. This process is done in every network. The sum of all messages, when a Dnode requests adjacencies, is shown in equation 3.8.

$$messages = 1 + 5 \cdot n \qquad (3.8)$$

3.2.4 Protocol description and recovery algorithms

When a Dnode joins the interconnection system, it sends a discovery message with its networkID to Onodes known in advance or by bootstrapping [72]. Only Onodes with the same networkID will reply with their λ parameter. Dnode will wait for a hold time and choose the Onode with highest λ. If there is no reply for a hold time, it will send a discovery message again. Next, Dnode sends a connection message to the elected Onode. This Onode will reply a welcome message with the nodeID assigned. Then, the Onode will add this information to its Dnodes' table (it has all Dnodes in its area). Onodes use this table to know Dnode's δ and λ parameters. Finally, Dnode will send keepalive messages periodically to the Onode. If an Onode does not receive a keepalive message from a Dnode for a dead time, it will erase this entry from its database. Steps explained can be seen in figure 3.1.

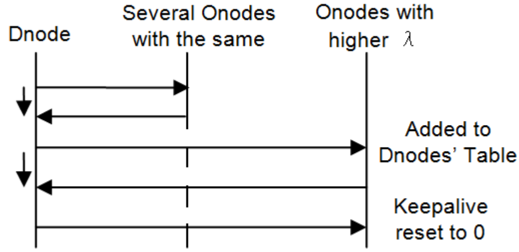

Figure 3.1. Messages when a new Dnode joins the architecture.

First time a Dnode establishes a connection with an Onode, it will send to that Onode a requesting message to establish connections with Dnodes from all other networks. This message contains destination networkID, or a 0xFF value (broadcast), sender's networkID, sender's nodeID and its network layer address. When the Onode receives the request, it will broadcast the message to all Onodes from other networks. The message is propagated using Reverse Path Forwarding Algorithm. Only Onodes from other networks will process the message. When an Onode receives this message, it will choose its two connected Dnodes with higher λ and it will send them a message informing they have to establish a connection with a Dnode from other network. This message contains the nodeID and the requesting Dnode's network layer address. When these Dnodes receive that message, they will send a message to connect with the Dnode from the first network. When the connection is done, Dnodes from the second network will send a message to its Onode to inform they have established a connection with the requesting Dnode. If Onode does not receive this message for a hold time, it will send a new message to the next Dnode with highest λ. This process will be repeated until Onode receives both confirmations. When the requesting Dnode receives these connection messages, it will add Dnode with highest λ as its first neighbor and the second one as the backup. If the requesting Dnode does not receive any connection from other Dnode for dead time, it will send a requesting message again. Then, both Dnodes will send keepalive messages periodically. If a Dnode does not receive a keepalive message from the other Dnode for a dead time, it will erase this entry from its database. Every time a Dnode receives a search or data transfer for other networks, it looks up its Dnodes' distribution table to know to which Dnode send the search or data. Figure 3.2 shows all steps explained.

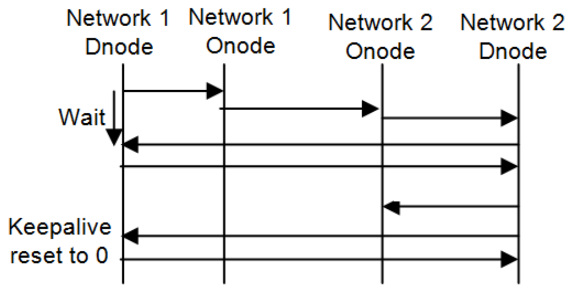

Figure 3.2. Messages to request Dnode's connections.

When an Onode receives a new networkID, it will send a message to all Dnodes in its zone. Subsequently, Dnodes will begin the process to request Dnodes from the new network. Dnode with highest δ will be the backup Onode and it will receive Onode backup data by incremental updates. This information is used in case of Onode failure. Onode sends keepalive messages to the backup Dnode periodically. When an Onode fails, backup Onode will check it because of the lack of keepalive messages for a dead time, so, it will start Onode functionalities using the backup information. When an Onode leaves the architecture voluntarily, it will send a message to the backup Dnode. The backup Dnode will send it an acknowledgement, and, when it receives the ack message, it will send a disconnection message to its neighbours and leave the architecture. Then, the Dnode that is the backup Onode starts Onode functionalities using the backup information.

When a Dnode leaves the architecture voluntarily or fails down, Onode and adjacent Dnodes will ckeck it because they do not receive a keepalive message for a hold time. They will delete this entry from their Dnodes' database and adjacent Dnodes will request a new Dnode for this network as explained in figure 3.2.

3.2.5 Messages Bandwidth

We have developed 31 messages for the interconnection system. They can be classified in 2 classes:

1) Fixed size messages.
2) Messages which size depends on the number of neighbours, the size of the topological database or the backup information (O neighbors, ODB, Backup O in table 3.2).

In order to know the bandwidth consumed by those messages, let's suppose they are sent using TCP/IP over Ethernet headers. The sum of those headers is 58 Bytes (as is has been considered for other P2P protocols [73]). We have considered that networkID, nodeID, λ and δ parameters use 32 bits and, in order to calculate messages that depends on other parameters, every Onode has 4 neighbors, Dnode's table has 28 entries and the topological database may use the available TCP/IP over Ethernet payload. On the other hand, the limit of the TCP/IP over Ethernet payload is 1460 Bytes. Table 3.2 shows messages classified taking into account their number of bits. Figure 3.3 shows their bandwidth (we have used the numbers shown in table 3.2).

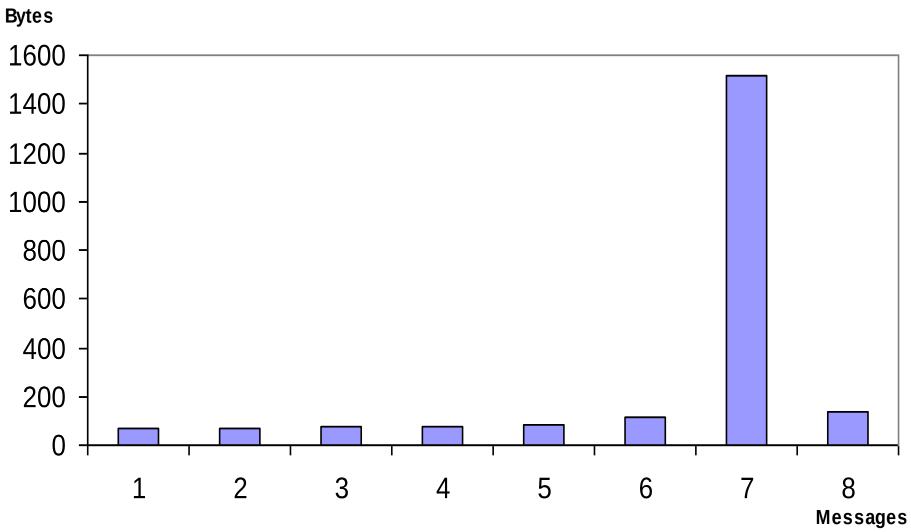

Figure 3.3. Messages bandwidth consumption

Table 3.2. Message classification.

Nº	Messages
1	D discovery, D connect y O request
2	D discovery ACK, Elected Dnode ACK, O discovery, Failed O, Failed O ACK, New O, Change nodeID, O disconnect, O conversion.
3	New Network, O discovery ACK, O connect.
4	Welcome D, Keepalive D, DDB Request, DD connect, Welcome DD, Keepalive DD, Keepalive O, O reply.
5	Elected Dnode, Welcome O, O replace.
6	O neighbors
7	ODB
8	Backup O

Messages with higher bandwidth are the ones that send the topological database and the backup information. Because of it, we have designed the system in such a way that these are only sent when changes take place. First time, both messages send the whole information, next times only updates are sent.

3.2.6 Experimental Measurements

We have developed a desktop application tool, using Java programming, to run and test the designed protocol. We have programmed Dnode and Onode functionalities. The application allows us to choose the multimedia network to be connected with. We can configure the type of files or data to download. We can vary some parameters for δ and λ calculation and the maximum number of connections, maximum CPU load, upstream and downstream bandwidth, keepalive time and so on.

We have used 24 Intel ® Celeron computers (2 GHz, 256 RAM) with Windows 2000 Professional © OS to know the protocol bandwidth consumption. They were connected to a Cisco Catalyst 2950T-24 Switch over 100BaseT link. 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. We have tested three scenarios:

1. First scenario has only one group. It has only one Onode and there are 23

Dnodes in the group. We began to take measurements before we started the Onode.

2. Second scenario has two groups. Every group has one Onode. There are 11 Dnodes in each group. We began to take measurements before we started 1st Onode. We started a Onode from 1st group, 20 seconds later we started a Onode from the 2nd group, 20 seconds later we started 11 Dnodes from the 1st group and 20 seconds later we started 11 Dnodes from the 2nd group.
3. Third scenario has three groups. Every group has one Onode. There are 7 Dnodes each group. We began to take measurements before we started the Onodes. We started a Onode from 1st group, 10 seconds later we started 7 Dnodes from the 1st group, 10 seconds later, we started a Onode from the 2nd group, 10 seconds later, we started 7 Dnodes from the 2nd group, 10 seconds later, we started a Onode from the 3rd group, finally, 10 seconds later, we started 7 Dnodes from the 3rd group.

Figure 3.4 shows the bandwidth consumed inside one group (1st scenario). There are some peaks because of the sum of joining discovery and keepalive messages (every 60 seconds) between Dnodes and the Onode. There are also DDB Request messages from Dnodes to other groups. Figure 3.5 shows the number of messages per second sent in the 1st scenario. Figure 3.6 shows the bandwidth consumed in the 2nd scenario. First peak (around 1 minute) shows when we joined 11 Dnodes from the 2nd group. Since this point, the number of octets in the network is higher because of keepalive messages. Figure 3.7 shows the number of messages per second sent in the 2nd scenario. Figure 3.8 shows the bandwidth consumed in the 3rd scenario. First peak (around 30 seconds) shows when we joined 7 Dnodes from the 2nd group. Once the 3rd group is started the bandwidth consumed is higher, but peaks are just bit higher than other scenarios. Figure 3.9 shows the number of messages per second in the 3rd scenario.

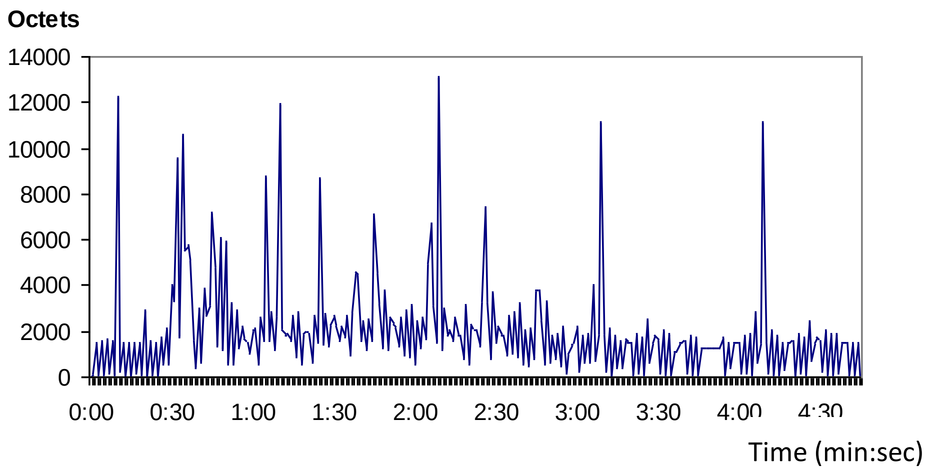

Figure 3.4. 1st scenario. Bandwidth.

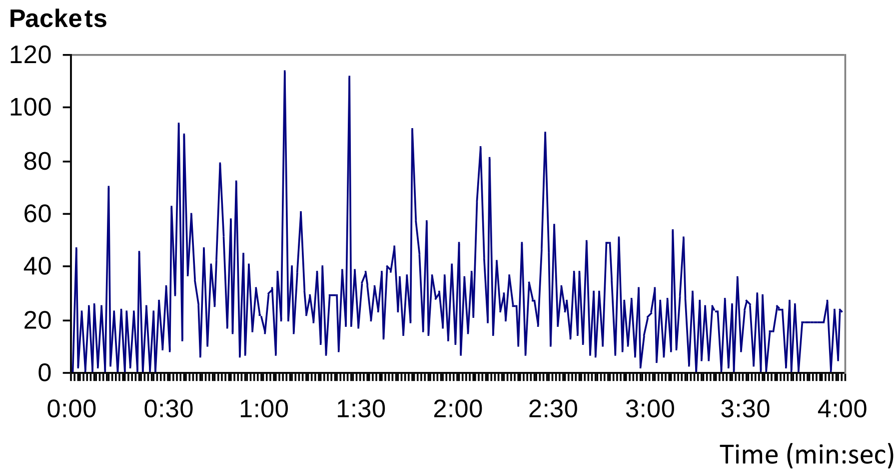

Figure 3.5. 1st scenario.Number of messages.

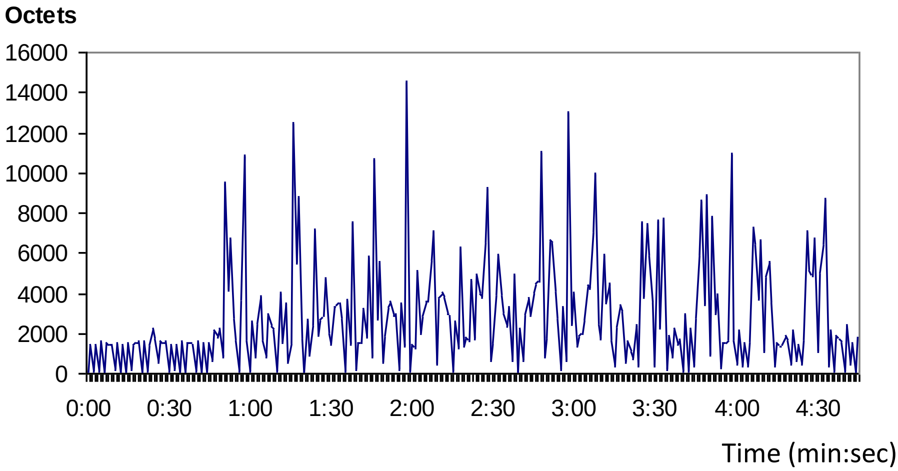

Figure 3.6. 2nd scenario. Bandwidth.

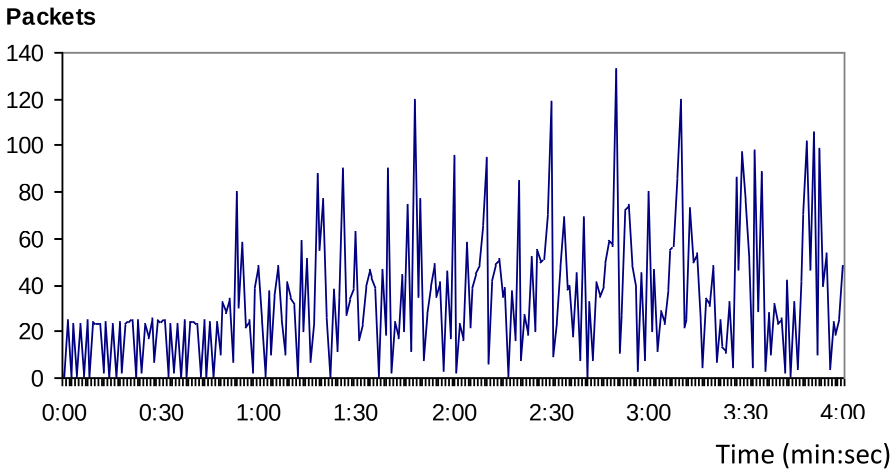

Figure 3.7. 2nd scenario. Number of messages.

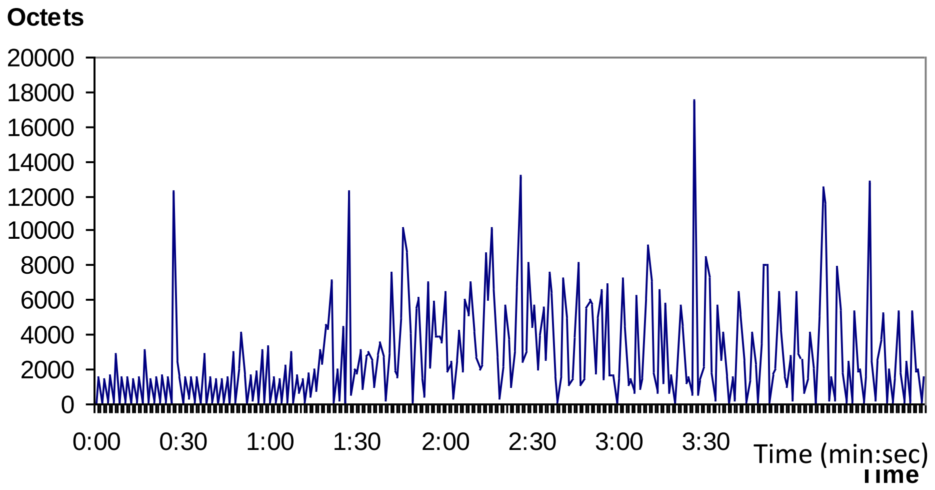

Figure 3.8. 3rd scenario. Bandwidth.

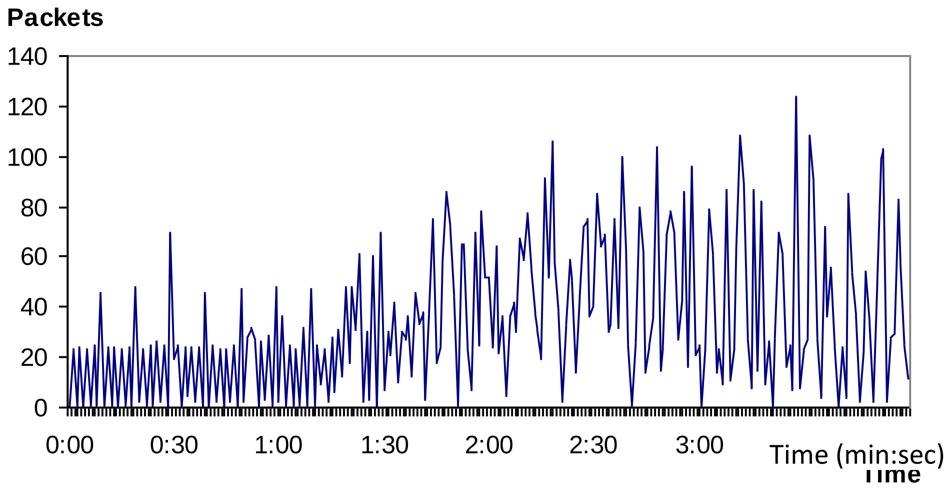

Figure 3.9. 3rd scenario. Number of messages.

3.3 A Group-based Content Delivery Network Arquitecture and Protocol

The idea behind Content Delivery Networks (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 CDN provides scalability, fault tolerance, and load balancing for the content delivery 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, they are also applied for delivering media streaming [74].

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 [63].

In the literature, several deployed CDNs that deliver many types of content over Internet can be found. Akamai [75] provides a distributed computing platform that delivers web, media streaming, software and applications. Another popular CDN is CODIS [76], 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 [77], 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 [78], that is being used to deliver media streaming. There are also general purpose open developments of CDNs such as Globule [79], Coral [80], CoDeeN [81] with different structure and operation protocols. A Model for Content Internetworking is published in the RFC 3466 [82]. 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 [14]. Hierarchical overlay network topology is required for a large-scale CDN to perform content delivery scalable and efficiently. Grouping nodes into clusters is one of the schemes most used when scalability is needed [16]. 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 CDN 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.

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 subsection 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 [83], 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 [13], 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 [84] [85].

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.

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.

3.3.1 Architecture description

From the logical point of view, the architecture is based on a two-layer model. The surrogates placed in the upper layer are called Control Surrogates (CS) and the surrogates in the lower layer are called Distribution Surrogates (DS). The CS controls 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 DSs in its group. The information between CSs is routed using SPF algorithm [86], 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 [69]. 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.

An example of the architecture proposed is shown in figure 3.10. 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).

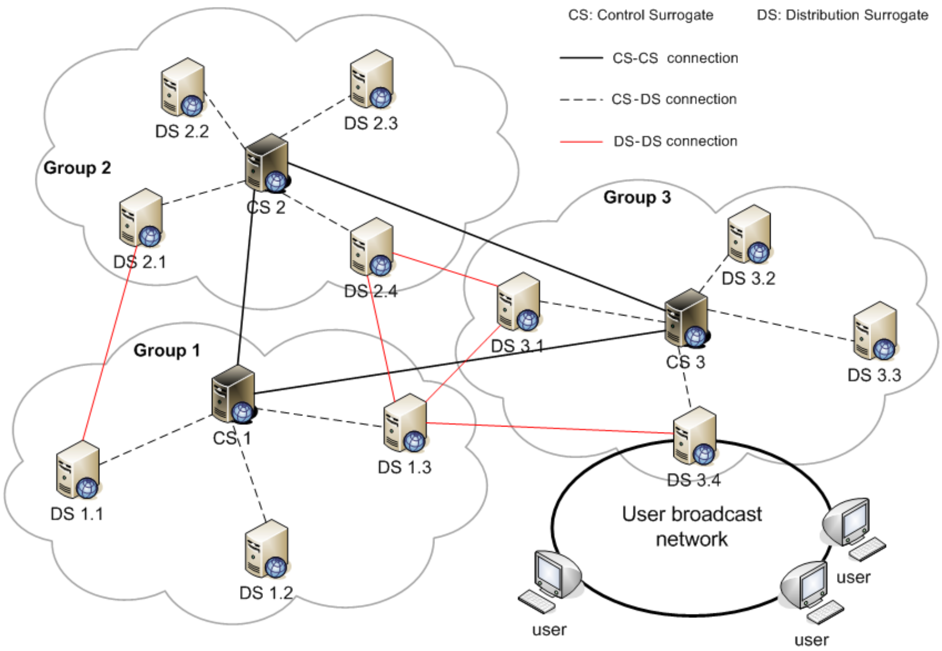

Figure 3.10. Proposed architecture topology example.

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.

Let P(i,j) be the network remoteness parameter between the *i*th and *j*th surrogates. It is defined in equation 3.9.

$$P(i,j)=\begin{cases}\alpha\cdot\dfrac{d(i,j)}{Max\_distance}+\beta\cdot\dfrac{RTT(i,j)}{Max\_RTT} & when \quad d(i,j)\le Max\_distance \;\; and \;\; RTT(i,j)\le Max\_RTT \\ \infty & otherwise\end{cases} \quad (3.9)$$

Where α and β are between 0 and 1, and allow assigning different weights to distance and RTT measurements. *d(i,j)* and *RTT(i,j)* are the distance and the Round Trip Time between *i*th and jth surrogates respectively.

Max_remoteness is the maximum value of the remoteness parameter and it is established for the whole network in order provide lower content distribution times and to assure Quality of Service (QoS). Now we can see that *P(i,j) ≤ Max_remoteness ≤ 2*.

We have defined λ as the surrogate's capacity. 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 for both, CSs and DSs, is defined by equation 3.10.

$$\lambda=\text{int}\left[\frac{BW_{up}+BW_{down}}{256}+1\right]\cdot Available\_Con\cdot(100-load) \quad (3.10)$$

There are two types of λ, one for CSs and another for DSs. If the number of connections achieves CS_max_con or DS_max_con, then λ=0. On the other hand, if the 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 have a table with the connected 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 remoteness and λ parameter for each entry.

3.3.2 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 [72]. 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. The flowchart of the explained procedure is shown in figure 3.11.

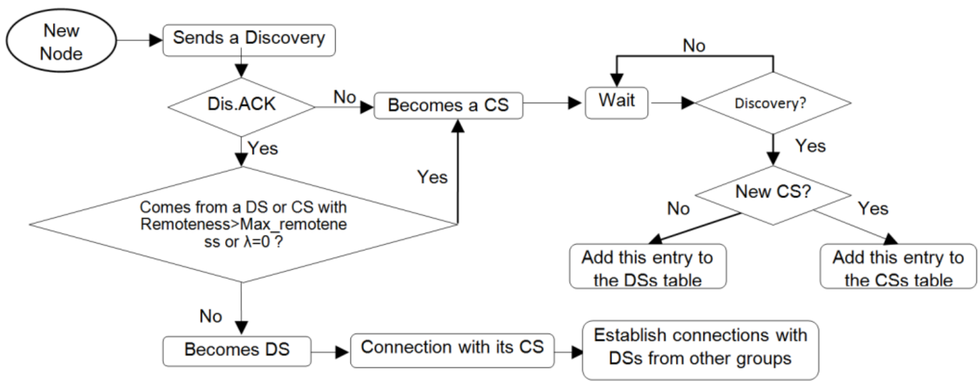

Figure 3.11. Protocol operation when there is a new node.

Steps followed by the protocol when a new surrogate joins the CDN and becomes a CS are shown in figure 3.12.

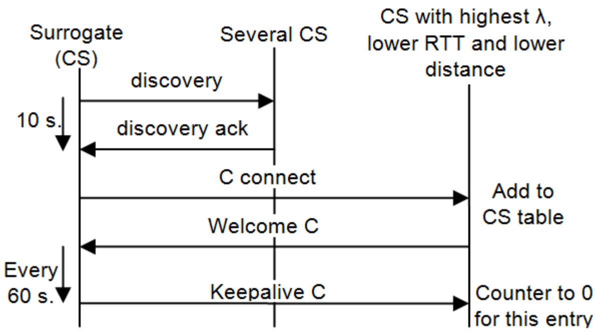

Figure 3.12. Protocol operation when there is a new CS.

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, which routes the packets based on the SPF tree.

Otherwise, if the surrogate receives a "discovery ack" reply from a CS which remoteness parameter is equal or lower than Max_remoteness or λ=0, the new surrogate will choose the best CS to have a connection with, taking into account λ parameter (higher λ values are preferred) and becomes a DS. The combination of the remoteness and λ parameters in the election could be 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 keepalive messages periodically to the CS 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 3.13 shows steps explained.

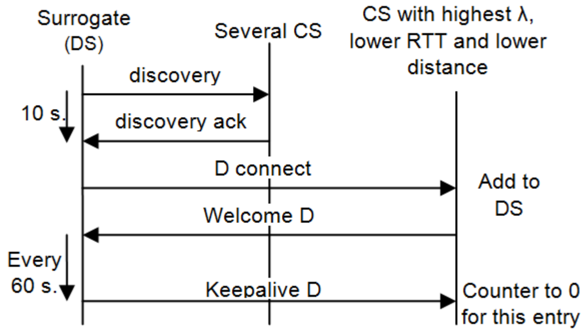

Figure 3.13. Protocol operation when there is a new DS.

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. It is routed through the CS’s network using the groupID. When a CS receives that message, it chooses the best DS using the closest position to the requester and the λ parameter of the DSs in the DS table. Then, the CS sends the “elected DS” message to the selected one and waits for an “elected DS ack” message to be advised that the DS knows it. 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 3.14.

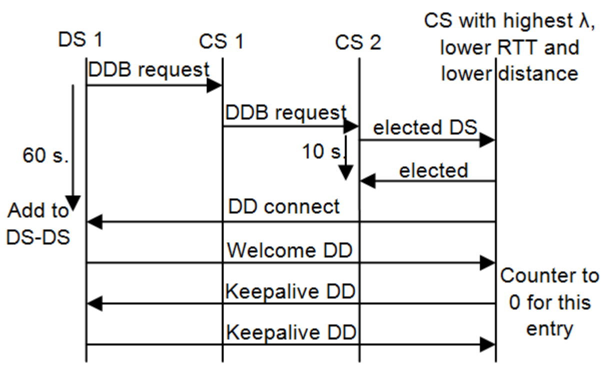

Figure 3.14. Protocol operation to establish a connection between DSs

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. It will also send a “DD disconnect” message to the connected DSs from other groups. They will erase this entry from their DS-DS table and will look for a new DS for that group. If the node fails down, the CS of its group and DSs from other groups will know it because of the missing of keepalive messages and they will erase that entry from their tables.

The DS with lowest remoteness parameter in the CS table is the backup CS. The backup CS has 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 neighbour CSs (and they will erase that entry from their CS table) and to the backup CS in its group. That update will be propagated through the CS network using the Reverse Path Forwarding algorithm. 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 3.15 shows the described procedure.

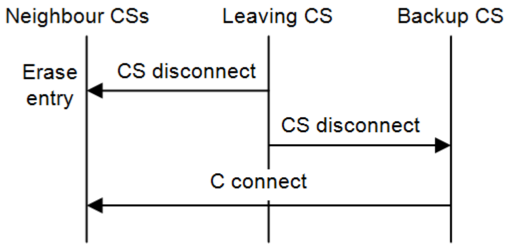

Figure 3.15. Protocol operation when a CS leaves the CDN.

If the CS fails down, it will be known because of the missing keepalive messages. 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.

3.3.3 Experimental Measurements

This section shows the real measurements taken from the control messages and the performance of the surrogates of a deployed CDN.

3.3.3.1 Test bench

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 that all other ports) to be able to capture data using a sniffer application. We selected only 14 surrogates in order to show a clear picture of the topology when the group-based CND is set up. In order to know the control traffic of the developed group-based Content Delivery Network, we have placed 14 surrogates in the positions shown in table 3.3. The values of the parameters are shown in table 3.4. As all surrogates were connected to the same switch, RTT values were the same so the distance was the decisive parameter to calculate the remoteness.

Table 3.3. 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

Table 3.4. The values of the parameters used to take measurements.

Parameters	Values
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

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 3.16. 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 using 2 dimensions, but it could be modified to 3 dimensions.

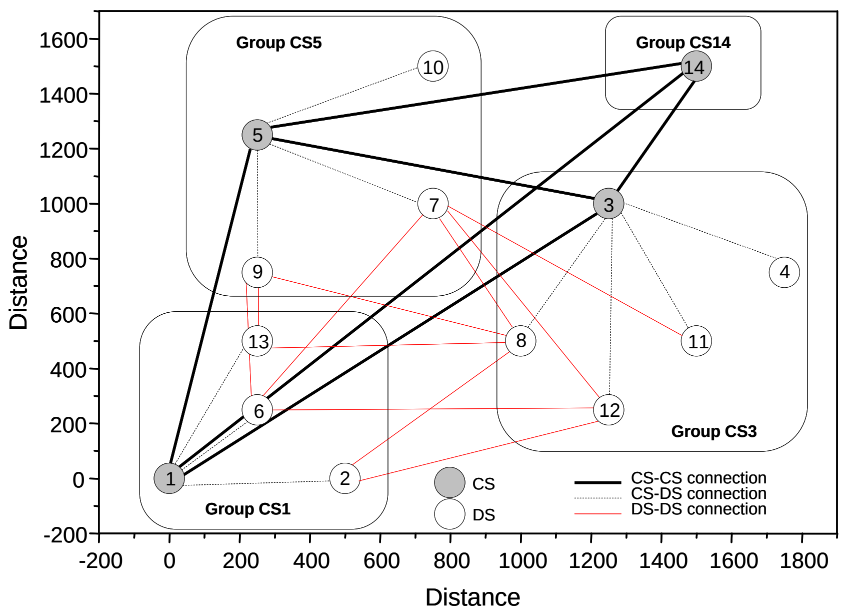

Figure 3.16. Surrogates distribution and connections established when the CDN has been set up.

3.3.3.2 Network 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.

We observed that when all these surrogates 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 3.17. 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.

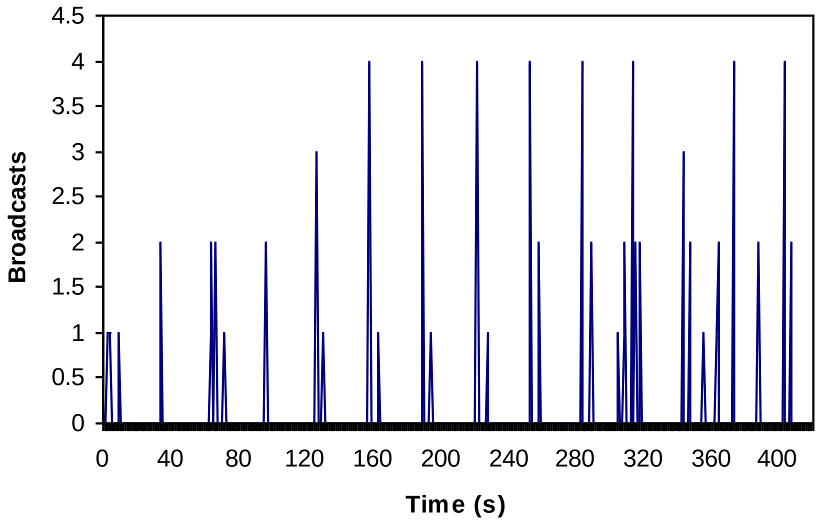

Figure 3.17. Broadcasts in initial process.

Figure 3.18 shows the total amount of control traffic introduced by our CDN. We can see that only 1024 Bytes/s are consumed when there are 14 surrogates. There are peaks every 60 seconds approximately because of the keepalive messages and the messages sent by the joining nodes in the initial process. When the network has converged, the control traffic is very similar to the initialization stage. It happens because the developed architecture does not send a high control traffic load when it starts in the initial phase. It allows us to demonstrate that it will introduce very little additional traffic when the topology changes.

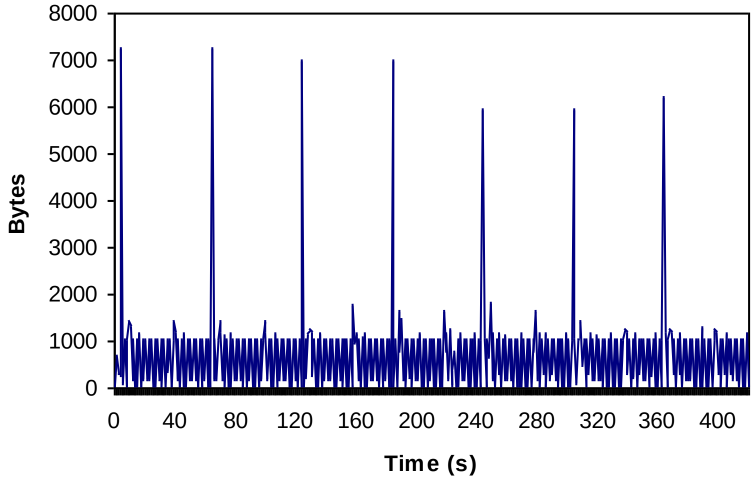

Figure 3.18. Control traffic in initial process (B/s).

We measured the number of packets in the whole network when no surrogate joins the network and we got an average value of 16 packets per second (see figure 3.19). There were peaks of around 34 packets per second. It was because of keepalive messages and the messages sent by new joining surrogates. Figure 3.20 shows the behavior of the CDN when the network has converged. We have observed that the number of broadcast decreased extremely. 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 surrogate in the network.

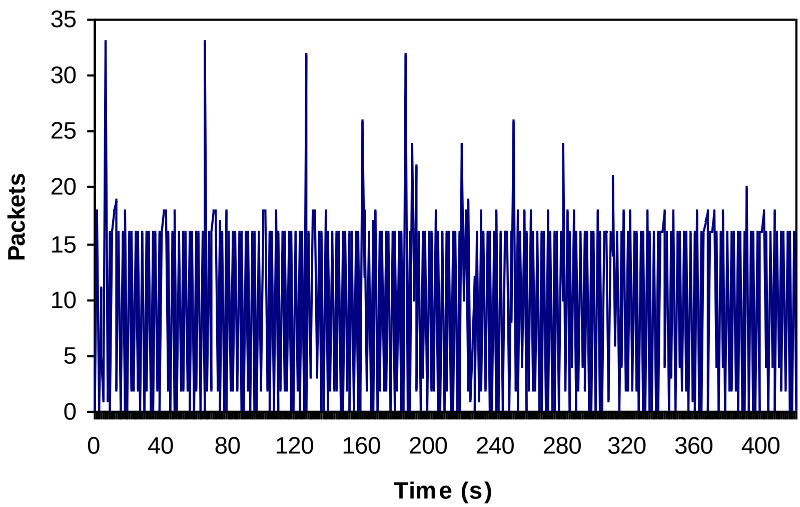

Figure 3.19. Control traffic in initial process.

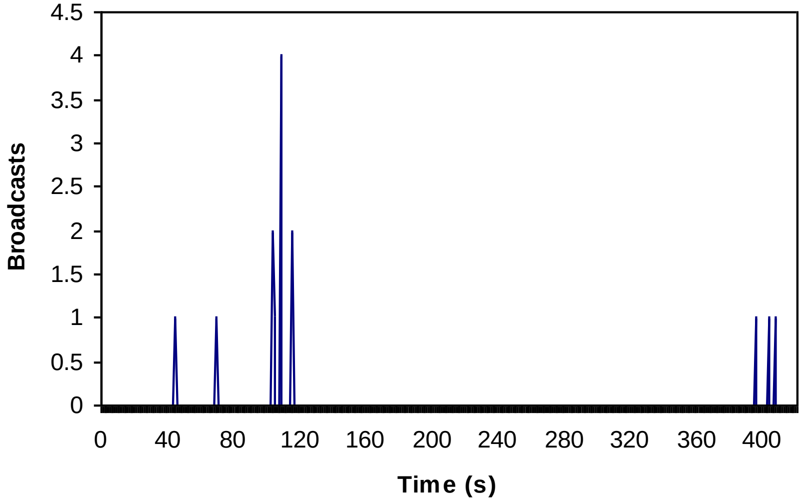

Figure 3.20. Broadcasts when the CDN has converged

3.3.3.3 Measurements of Scalability

In order to study the scalability of our proposal, we considered two scenarios. The first one was formed by 11 surrogates into 3 groups (first group had 4 surrogates, the second group had 4 surrogates and the third group had 3 surrogates). In the second scenario we duplicated the number of surrogates in each group (first group had 8 surrogates, the second group had 8 surrogates and the third group had 6 surrogates). We do not present the topology of both scenarios because it do not affect to the measured traffic. Notice that this is a real scenario in a laboratory, not a simulation, so it is difficult to measure 100 or 1000 surrogates in that environment. The values of the parameters were the same as the ones used for the network measurements (shown in table 3.4). As we did in the previous subsection, we started the nodes in a sequentially order every 30 seconds. Measurements were taken during 690 seconds in order to assure to show the convergence of the network. Figures 3.21, 3.22 and 3.23 show a comparison of how evolves the network until it converges for 11 surrogates vs. 22 surrogates.

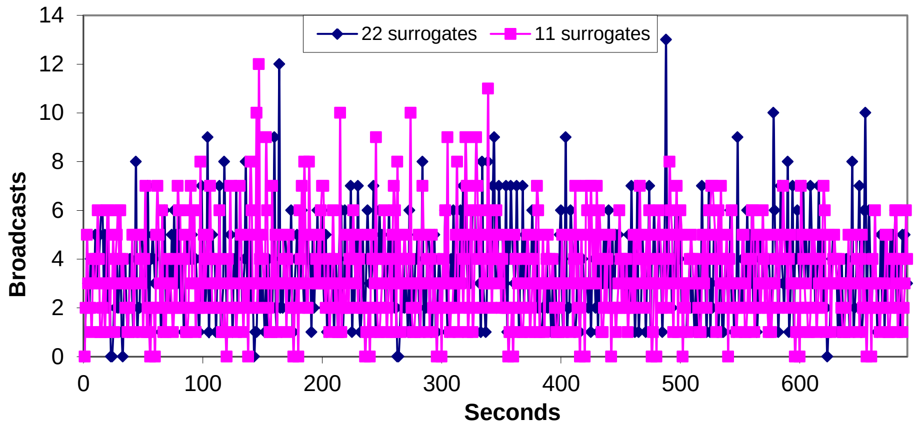

Figure 3.21. Network convergence. Number of broadcasts/s.

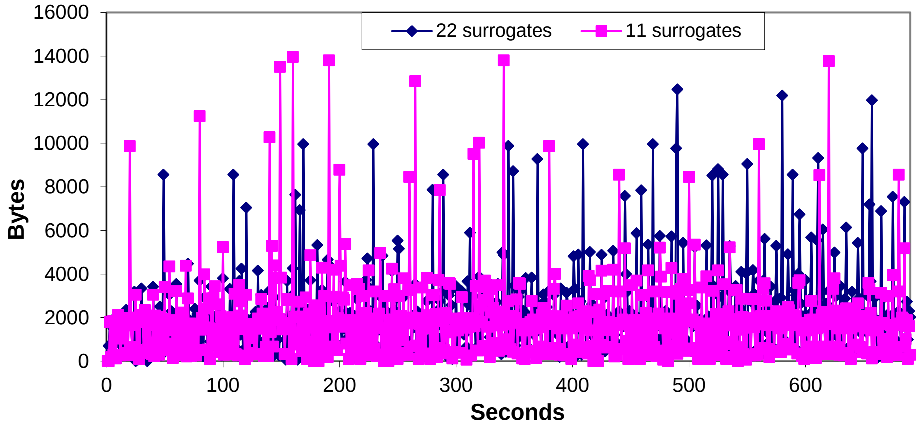

Figure 3.22. Network convergence. Bytes/s. c) Packets/s.

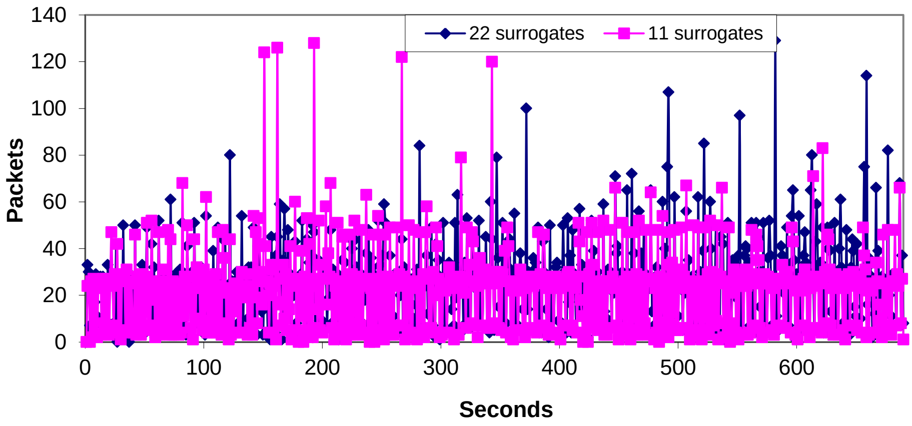

Figure 3.23. Network convergence. Packets/s.

Figure 3.21 shows that both 11 surrogates and 22 surrogates have almost the same broadcasts/s (there is an average of 3.40 for 11 surrogates and 3.44 for 22 surrogates). On the other hand, there has been a maximum value of 12 broadcasts/s for 11 surrogates and a maximum value of 13 broadcasts/s for 22 surrogates. Figure 3.22 shows that there are more Bytes/s in the 22 surrogates' scenario (there is an average value of 2149.11 Bytes/s) than in the 11 surrogates' scenario (there is an average value of 1849.21 Bytes/s). We have duplicated the number of surrogates, but the number of Bytes/s has been incremented only 16.22 Bytes/s. Figure 3.23 shows that there are very few more packets in the 22 surrogates' scenario (an average of 25.1 packets/s with a maximum value of 128 packets/s) than in the 11 surrogates' scenario (an average of 22.3 packets/s with a maximum value of 129 packets/s).

3.3.3.4 Surrogate measurements

In order to obtain measurements from the surrogates, we considered four scenarios from figure 3.16. 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 surrogates 11, 9 and 2. Surrogate 2 was distributing video to surrogates 8 and 12 while it was sending video streams to its user's network. Surrogate 11 was distributing video to surrogate 7 and to its users. Surrogate 9 was receiving video from surrogate 13. The fourth scenario was given by surrogate 12. We compared what happens in a surrogate when it receives different sizes of files.

Figure 3.24 shows the % of processing of surrogate 11 in scenario 1. It was distributing 200 MB while it was sending video streams to its user's network at the same time. We 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 was 20.8%. This variation is because 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 incremented the size of the content distributed in the CDN in the second scenario. Figure 3.25 shows the % of processing time measured when it is being distributed 800 MB. The time needed to transmit the file is greater than first scenario. 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 was distributed to other surrogates and 32.82 % when the surrogate sends video streams to its user network. The % of processing time is higher (20.79%) when it is receiving than when it is sending.

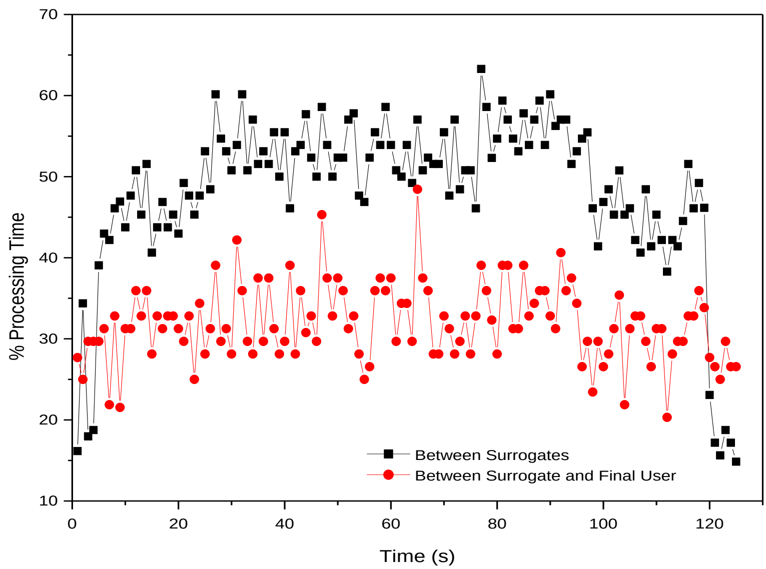

Figure 3.24. % processing time when it is distributed 200 MB.

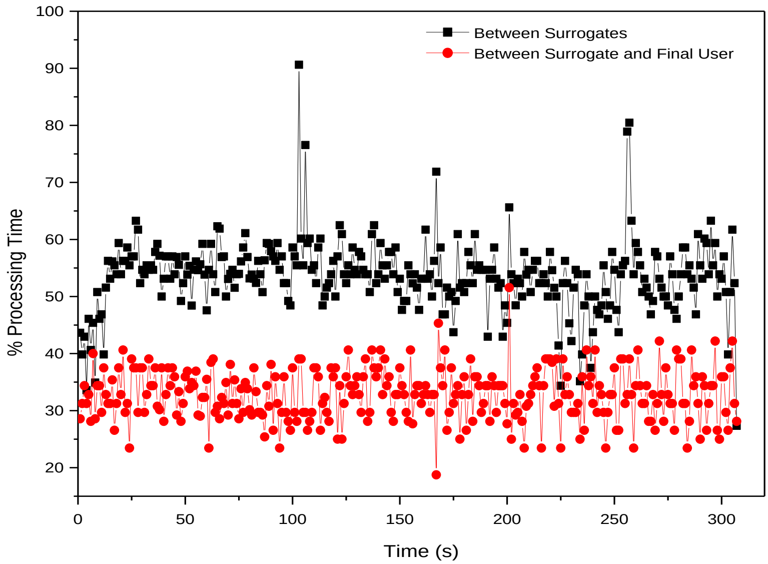

Figure 3.25. % processing time when it is distributed 800 MB.

In figure 3.26 the measurements taken from scenario 3 are shown. We compared the number of readings of the cache for surrogates 11, 9 and 2 when the file was 1 GB. The number of readings in cache per second was around 330 for surrogate 2. Surrogate 11 had 91 readings of cache per second in average. Surrogate 9 had around 6 readings of cache per second. We think that this low value 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 surrogates while it was sending video streams to its user's network. Figure 3.27 shows the number of packets per second received by surrogate 12 when it was receiving three sizes of files (200 MB, 500 MB and 800 MB). It shows that higher size files do not imply larger times. In order to transmit 800 MB, we have obtained 2.59 times more than the time needed for 200 MB. On the other hand, we have observed that when higher files are transmitted, the graph obtained for the received packets per second are less stable than for smaller files.

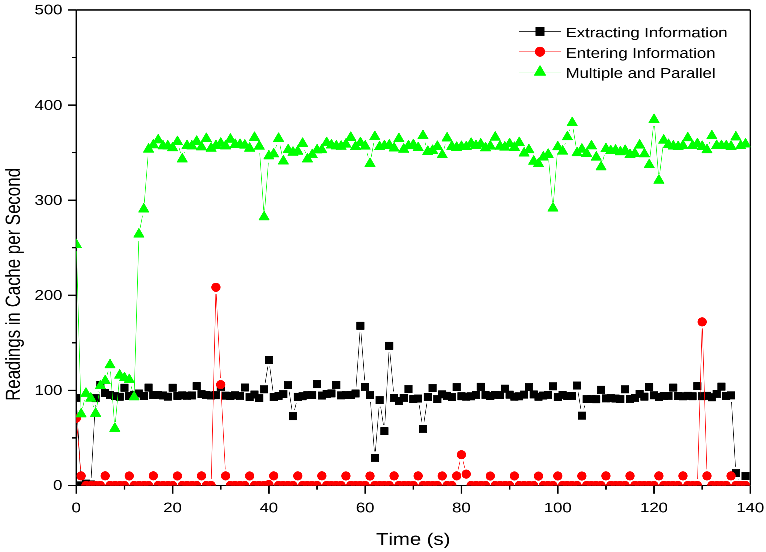

Figure 3.26. Readings in cache per second when it is distributed 1 GB

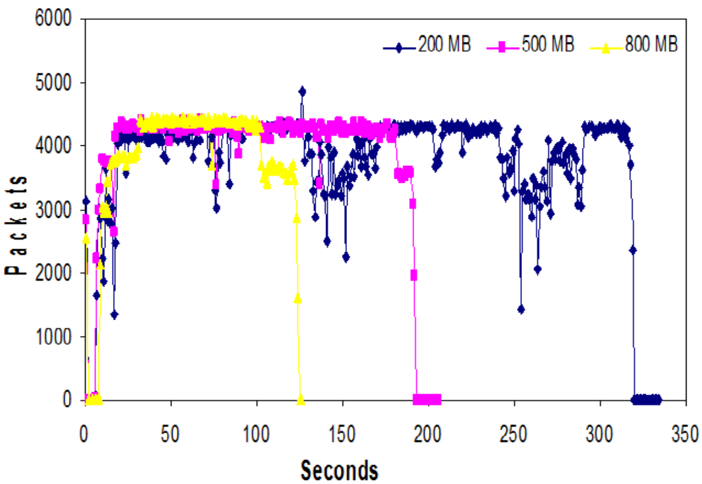

Figure 3.27. Packets per second received for different video files.

3.4 A Cluster-Based Architecture to Structure the Topology of Parallel Wireless Sensor Networks

In cluster based architectures, mobile nodes are divided into virtual groups. Each cluster has adjacencies with 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 [87]. Figure 3.28 shows an example of a topology with two clusters where there are two cluster head nodes, several cluster members and two cluster gateways. 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. Inter cluster data transfer takes place through the cluster gateways [88]. Cluster members are the rest of the nodes in a cluster. In this kind of network, Cluster Head nodes are used to control 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, only cluster gateways.

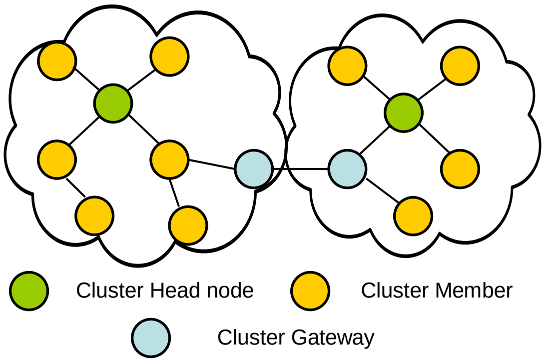

Figure 3.28. Cluster topology example.

There are many cluster based architectures [18]. Sensor networks clustering schemes can be classified according to several criteria. E.g., they can be classified according to whether the architectures are based on Cluster Head [50] or on Non Cluster Head [51]. The first architecture needs Cluster Head to control and manage the group, and the second one does not have a specific node to perform this task. Another way to differentiate the cluster-based architectures is observing the hop distance between node pairs in a cluster. The schedules can be divided into 1-hop clustering [57], multi-hop clustering [89] or multilevel clustering [90]. The maintenance of the hierarchical multilevel requires heavy communication overheads due to random change of multilevel topology. By contrast, the cluster head of single level clustering is simple, since it only tracks local topology changes due to host mobility.

In addition to these classification criteria, reference [18] presents another classification based on the objectives of the clustering protocols. There are six clustering schemes: dominating-set-based (DS-based) clustering [91], low-maintenance clustering [50], mobility-aware clustering [92], energy-efficient clustering [51] [93], load-balancing clustering [94] and combined-metrics based clustering [57].

The clustering architectures provide many benefits. Reference [53] shows the most important features of cluster-based architectures over ad-hoc and sensor networks. The last feature is strongly linked with energy conservation, given that clustered wireless sensor networks offer two major advantages over their non-clustered counterparts; firstly, clustered wireless sensor networks are capable of reducing the volume of inter-node communication by localizing data transmission within the formed clusters and decreasing the number of transmissions to the sink node; secondly, clustered wireless sensor networks are capable of extending the nodes' sleep times by allowing cluster heads to coordinate and optimize the activities.

In the literature several cluster deployments and proposals can be found. One of them is the one presented in [95], which proposes a cluster-based system to overcome problems of bottleneck and poor scalability. In [96], Yi Zou and Krishnendu Chakrabarty proposed a cluster-based distributed sensor network deployment and target localization to enhance the coverage after an initial random placement of sensors.

All these works use a routing protocol inside the cluster ([97] [98]) or they use the cluster to build a unique protocol for the entire network (such as the one presented by Keun-Ho Lee et al. in [99], where the cluster is used for authentication), but none of the protocols seen use cluster-based schemes to build different networks. So, this is the first time that different sensor networks have been built using cluster-based schemes.

In this subsection we present a proposal where nodes from different clusters have to communicate in order to build different wireless sensor networks (such as virtual networks). In order to control and manage the system, some limitation parameters have been added. Connections will be established if they are close enough and only if they are the same type of sensor. The main contribution of our work is the design and verification of a new protocol and its comparison with other protocols in existence.

3.4.1 Problem Formulation and Application Environments

Let us suppose an environment where a great variety of sensors must be scattered to take measurements from the environment or the same type of sensor but with different types of profiles. Let us suppose that the whole area is divided into zones and each zone has one or several types of sensors (humidity, temperature, wind, movement, etc.) organized by a central node as a cluster. Let us also suppose that exclusive networks with the same type of sensors (network of temperature sensors, network of wind sensors, network of humidity sensors, etc.) are needed. An example could be the use of clusters of different sensors for each tree in a forest. So, there will be as many networks as types of sensors. It could be also used to create virtual wireless sensor networks (the same concept as Virtual Local Area Networks in wired networks, VLANs, where the devices of the network are joined in logical independent topologies, although they are in the same physical topology).

Some examples given to explain the form of that exclusive network between nodes from different clusters are the following:

- They could have a different transceiver to connect to other cluster nodes but the same transceiver to connect to the cluster head node.
- They could use a different wireless protocol to connect to other cluster nodes but the same transceiver to connect to the cluster head node.
- They could be using different types of technology to connect to other cluster nodes but the same transceiver to connect to the cluster head node.

- They could be transmitting different types of data that is not understandable by other types of nodes, only by the same type of node and the cluster head node.
- They could have different types of profiles.
- They could be different types of devices.

Moreover, there are some statements that must be added:

- When a new sensor joins the architecture, it will belong to the zone of its nearest central cluster sensor.
- Due to processing consumption issues, the number of connections to the central cluster sensor should be limited, so when it reaches the maximum number of connections, the new sensor has to create a new cluster.
- Sensors will have connections only with the same type of sensors of neighboring groups in a predefined distance or coverage area, but not with nodes from other groups that are not neighbors.
- For energy saving purposes, when there are several sensors from other clusters in the sensor's coverage area, the one with higher capacity (which depends on the energy between other parameters that will be presented later) will be chosen as a neighbor.
- The network formed by sensors of the same type will have its own routing protocol algorithm.

Taking into account the aforementioned premises, several application environments can be found. Some of them are the following:

- It could be used in any kind of system where an event or alarm is based on what is happening in a specific zone, but conditioned to the events that are happening in neighboring zones. One example is a group-based system to measure the environmental impact of a place (forest, marine reef, etc.). It could be better measured if the measurements are taken from the plants and from the trees in that place with different type of sensors. Each kind of measurement could be taken from different groups of sensors, but those groups of sensors have to be connected with the same type of sensors in order to estimate the whole environmental impact.

- It could be used in body area sensor networks. The devices used to sense the body could be several types of sensor (pulse sensors, skin sensors, sweat sensors, etc.). A sensor may need to be connected with the same type of sensors of other zones of the body to form their specialized network in order to check the measurements of a specific parameter.
- It could be used to build networks of sensors with the same profile that come from different communities (each community will be a cluster).
- It could be used to build networks whose cluster can be formed by different types of devices such as mobiles, PDAs, PCs, sensors, but the requirement is for networks formed by the same type of devices.
- It could be used to for virtual wireless sensor networks creation.

3.4.2 Architecture Proposal Description

From the logical point of view, our proposal is based on a two-layer model, the organization layer and the distribution layer. All clusters must have both layers. Sensor devices (henceforth referred to as “nodes”) in the organization layer are called Cluster Heads (CH). Although they have capacities of sensing, they are the ones with higher capabilities of the cluster (how they are elected is defined later). The distribution layer is formed by cluster member nodes and cluster gateway nodes, henceforth called CMs. CHs also have CM capabilities. The same physical and MAC layers are used between CHs and between CMs and their CHs. CMs could use the same or have another type of physical and MAC layers, using different wireless transceivers, for their exclusive network [100]. CHs organize and control the CMs in their cluster and all CMs have to establish a connection with a CH to join its cluster. This connection can be established only if the distance between them is shorter than or equal to a predefined value. In the rest of the subsection, the distance will be considered as one of the limiting parameters to establish connections, but it can be changed by the Received Signal Strength Indication (RSSI) value or by the Signal Noise Ratio (SNR) value or by any other parameter that could be used to know if the candidate neighbor is reachable. In our design only one CH per cluster has been provided, but more could be added for scalability purposes.

CHs have connections with some CHs of others clusters. OLSR [101] has been chosen as the routing protocol to route information between CHs, but it could be changed by any other routing algorithm such as AODV [102], DSR [103] or TORA [104]. The organization layer is used to organize connections between CMs of different clusters. There are several types of CMs in a cluster depending on what they are sensing, or the profile, or the type of device. CMs have connections with the same type CMs of other clusters only if the distance between them is shorter than or equal to a predefined value. CHs can be any type of ad-hoc device or sensor. They can communicate with CMs of other clusters because they are CMs but only if they are the same type of CMs.

The number of clusters in the network is determined by the extension that is to be covered by the whole network. If a new zone needs to be covered, a new cluster has to be added. Although many types of sensors or types of devices can be added to any cluster, the application of the 20/80 rule (20% of CHs, 80% of CMs) is suggested [105].

An example of the architecture proposed is shown in figure 3.29. Although a CH is in both layers, they have been placed just in the organization layer to clarify the Figure. CHs have connections with some CHs from other clusters (lines formed by black points). All CMs have a connection with the CH of its cluster (lines formed by red points) and with the selected CMs from other clusters (solid black lines).

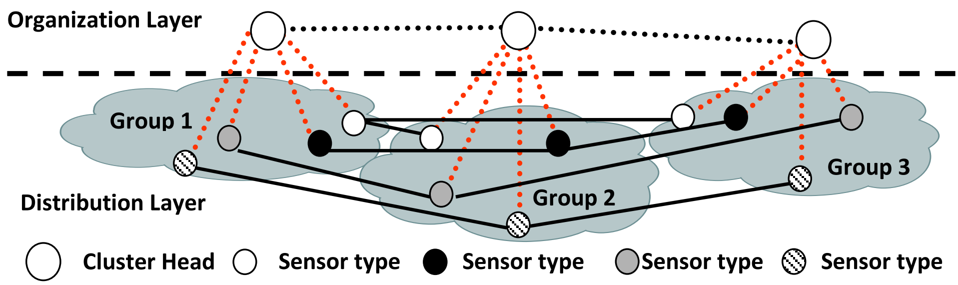

Figure 3.29. Proposed architecture topology example.

3.4.2.1 Identifiers and Predefined Parameters

Every cluster has an identifier called *clusterID*. When a new node joins a cluster it acquires a unique 32-bit node identifier called *nodeID* from the CH. The first node in a cluster will be the CH and will have *nodeID* = 0x01. Once the CH of the new cluster has contacted with other CHs in the network, it will acquire the first available *clusterID* and, then, it will try to connect with the same type of CMs from other clusters. All nodes in a cluster have the same *clusterID*. Any new node will join the cluster whose CH is closest.

Every new node must have the following parameters to join the proposed architecture:

- **Max_con:** Maximum number of supported connections from other nodes of the distribution layer.
- **Type:** It identifies the type of node.
- **Max_distance:** It is the maximum distance to be a neighbor. It is always shorter than or equal to the coverage area radius. It can be changed by the Received Signal Strength Indication (RSSI) value or by the Signal Noise Ratio (SNR) value. It is applied only to establish connections between CHs and their CMs and between CMs, but not between CHs because CH must have as many connections with other CHs as possible.
- **Position:** It could be given manually or by GPS.

It will have other parameters that vary along its existence in the architecture:

- Available_con: Number of available connections with other nodes of the distribution layer.
- E: % of energy consumption.
- L: % of available load. The load is the quantity of tasks the node is able to carry out at one time.

Two parameters have been defined to be used for the operation of the architecture.

3.4.2.2 δ Parameter

It depends on the node available energy and its age in the system (the lower *nodeID*, the older the node is). It is used to ascertain which node is the best one to be a Cluster Head node. This would seem to be anomalous since the oldest node should be the lowest energy node, but this parameter appears to consolidate the most stable nodes as the CHs (new ones could be mobile nodes or even with lower energy). So, when those in the cluster have low energy, only new nodes with very high available energy will have preference. A node with higher available energy and older will have higher δ. Equation 3.11 gives δ parameter.

$$\delta = (32 - age)\sqrt{1 - \frac{E^2}{K_1}} \quad (3.11)$$

Where *age* = $\log_2$(*nodeID*), so age varies from 0 to 31. E is the consumed energy. Its values vary from 0% to 100%. E = 0 indicates it is fully charged and E = 100 indicates it is fully discharged, that is, all energy is consumed. K_1 defines the minimum value of energy remaining in a node to be suitable for being selected as a neighbour. Figure 3.30 shows δ parameter values as a function of the node age for different available energy values. We have fixed $K_1=10^4$ to avoid having negative values inside the square root while it fixes the value of the square root between 1 and 0.

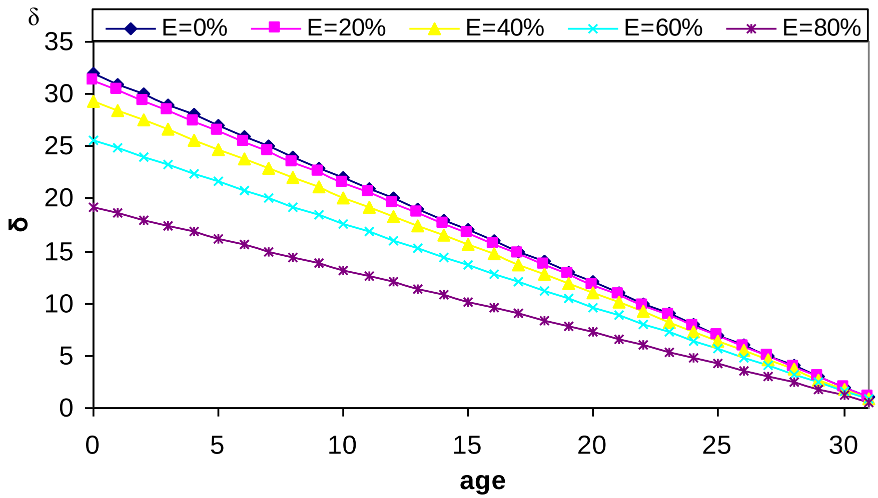

Figure 3.30. δ values as a function of node age.

3.4.2.3 λ Parameter

It is the capacity of a node. It is used by the CMs to determine the best CM to connect with when there are several choices. λ parameter depends on the node's number of available connections (*Available_Con*), its maximum number of connections (*Max_Con*), its % of available load (*L*) and its % of available energy (*E*). It is defined by equation 3.12.

$$\lambda = \frac{Available\_Con \cdot L + K_2}{Max\_Con} \cdot \sqrt{1 - \frac{E^2}{K_1}} \qquad (3.12)$$

Where 0 ≤ *Available_Con* ≤ *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 node. An energy consumption of 0 indicates it is fully charged and a value of 100 indicates that it is fully discharged. K_1 is defined as it was for δ parameter and K_2 gives λ values different from 0 in case of *L* = 0 or *Available_Con* = 0. The root is excluded from the division because when the node is fully discharged, λ parameter has to be 0. We have considered K_2 = 100 when *Available_Con* have low values (lower than 10) because it avoids having too high (or too low, depending on the case) λ values (in the range of]0,110]). Figure 3.31 shows λ parameter values when the maximum number of links for a node is 8 and all have the same available number of links (*Available_Con* = 4) as a function of the node available energy for different load values. It shows that as the Energy is being consumed, λ parameter is decreasing, but when it receives 80% of consumption, it decreases drastically, so the node is more likely to be chosen as a neighbor, in case of more available energy. Figure 3.31 also shows that a node with higher bandwidth is preferred.

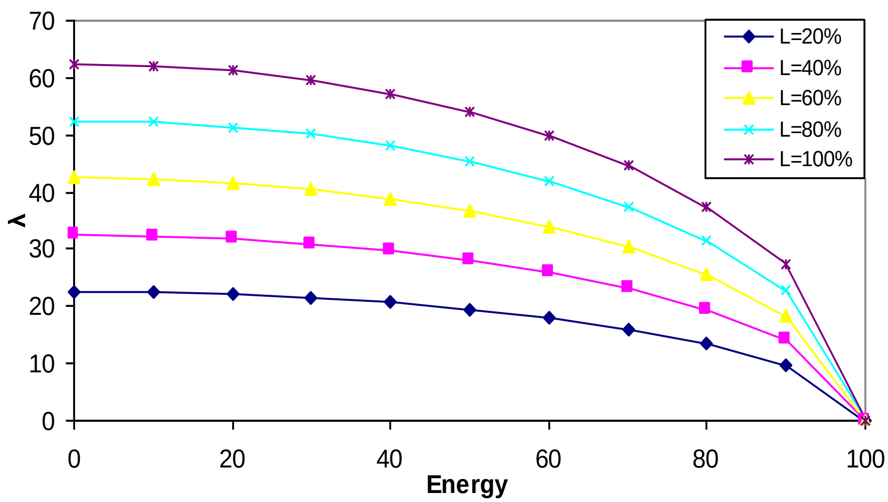

Figure 3.31. λ values as a function node Energy.

3.4.3 Scalability

It is known that cluster based systems are more scalable than other systems. This section shows why our proposal scales better than other proposals. First, we have to take into account that computation is much cheaper than communication in terms of energy dissipation [106]. So, what is desired is architecture with fewer retransmissions. This will imply a saving in energy of the whole system and it will give more scalability to the architecture.

Let a network of nodes G = (V, E) be, where V is the set of nodes and E is the set of connections between nodes. Let k be a finite number of disjoint clusters of V, so V = $\cup(V_k)$ and there is no node in two or more subsets ($\cap V_k = 0$), i.e., there are not overlapping nodes. Let us suppose N = |V| (the number of nodes of V) uniformly distributed in a region. Let us suppose that there is just one cluster head node per cluster, so there are k head clusters in the whole network. Equation 3.13 gives the number of nodes.

$$N = \sum_{i=1}^{k} |V_k| \qquad (3.13)$$

And the average number of neighbors of a cluster head will be given by equation 3.14.

$$Average = \frac{N}{k} - 1 \qquad (3.14)$$

Four main types of cluster architectures can be distinguished:

- 1-level cluster (see Figure 3.32). Cluster head nodes are connected directly without any intermediate node. The information between clusters is sent through the cluster head nodes.

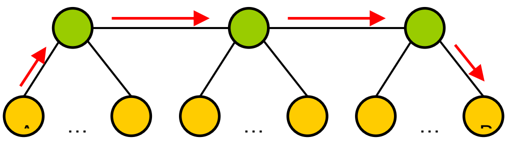

Figure 3.32. 1-level cluster.

- P-level cluster (see Figure 3.33). It is a hierarchical structure where there are different levels that connect the cluster head nodes. The information between clusters is sent through the hierarchy.

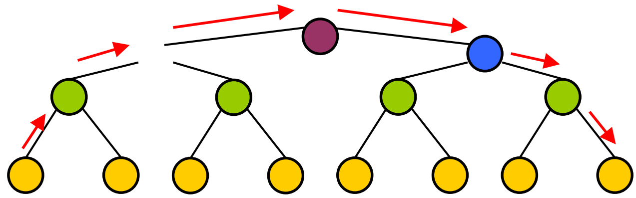

Figure 3.33. P-level cluster.

- Planar cluster with 1 hop (see Figure 3.34). The information between clusters is sent through the gateway nodes (not through the cluster head nodes).

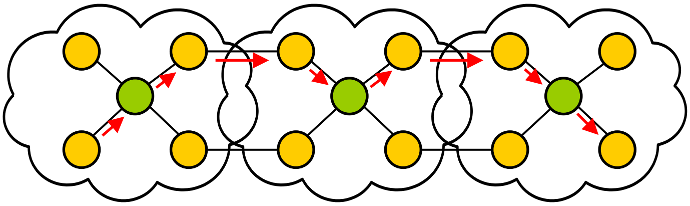

Figure 3.34. Planar cluster.

- Our proposal (see Figure 3.35). It joins nodes from different clusters. Two nodes could be connected directly (e.g. A➔B in figure 3.35), but we have considered a worst case where the communication has to be done through an intermediate node from other cluster.

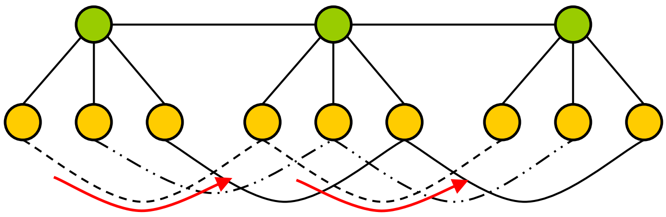

Figure 3.35. Our proposal.

The diameter of a network (d) is defined as the length of the delay-optimal path between the two farthest nodes. Figures 3.32, 3.33, 3.34 and 3.35 also show the path between the two farthest nodes. Table 3.5 shows the diameter for each type of cluster.

Table 3.5. Diameter for each type of cluster (Note: k is the number of clusters and P is the number of levels of the hierarchy)

Type of cluster	**Diameter (*d*)**
1-level cluster	$k+1$
P-level cluster	$2\cdot p$
Planar cluster with one hop	$3\cdot k-1$
Our proposal	$k-1$

Figure 3.36 shows the diameter of each cluster architecture (we have considered K = P for the P-level cluster).

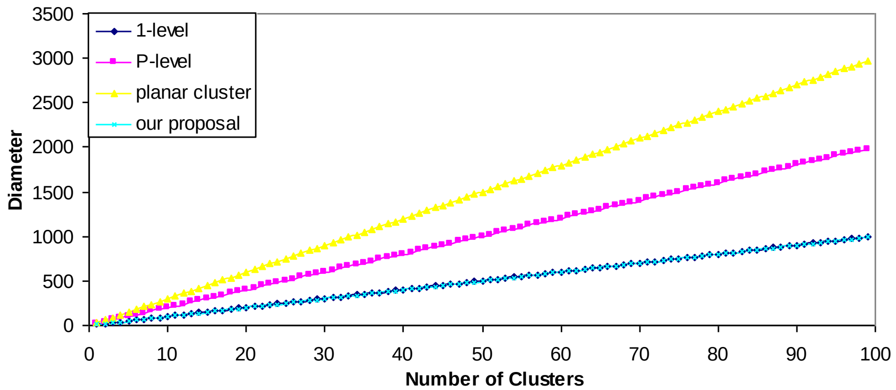

Figure 3.36. Diameter.

Figure 3.36 demonstrates that fewer hops are needed in our architecture than for the others (it has very few difference with the 1-level architecture), so the total routing overhead of the network is reduced and fewer retransmissions are needed, thus more energy is saved. This implies that our proposal scales better than the others.

3.4.4 Architecture Proposal Operation and Fault Tolerance

In order to join the architecture, the new node broadcasts a "Discovery" message. CHs will reply with a "Discovery ACK" message with their position and λ parameter. There could be 3 possibilities:

1. If it does not receive any reply in 10 seconds, it becomes a CH, so it creates the cluster and waits for new nodes. Ten seconds have been chosen because it is enough time to receive a reply from a near node. Later replies will be from nodes which are either too far or too busy.

2. If it receives some replies, but none of them are at a distance lower than the Max_distance, it becomes a CH and sends an "H connect" message to establish connections with selected CHs (based on their λ parameter). If the other CH confirms that connection, it adds this entry to its CH table and sends a "Welcome H" message with the last *clusterID* in the network. The CH will choose the next available *clusterID* for its cluster. Then, the new CH will send "Keepalive H" messages periodically (with its *clustered*) to its neighbor CHs in order to indicate it is alive. If a CH does not receive a "Keepalive H" message from a neighbor CH for a dead time, it would erase that entry from its neighbor database. "Keepalive H" messages contain sender's *clusterID* and λ parameter. The CH also follows the new CM process described later. Messages sent in this case, when there is a new CH, are shown in Figure 3.37. Once the discovering process has finished CH node's network works as a regular OLSR network.

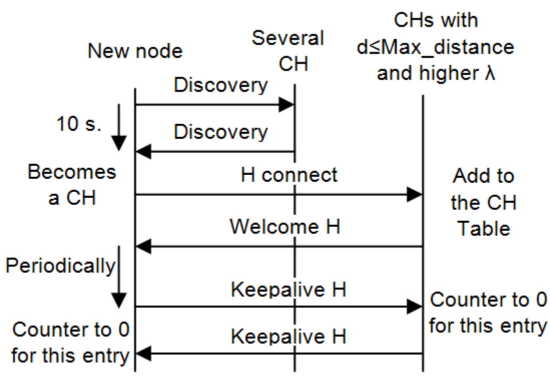

Figure 3.37. Protocol operation for a new CH.

If it receives one or several replies and all of them are within a distance lower than the Max_distance, it chooses the closest CH and in case of a draw, the one with highest λ parameter. Then, it sends an “M connect” message to establish a connection with the selected CH. If the other CH confirms that connection because it has not reached the maximum number of connections, it adds this entry to its CM table, sends a “Welcome M” message and the new node becomes a CM. If the CH does not agree the connection, the new node sends an “M connect” message to the second best CH and follows the same steps. This process is repeated until the new node reaches the last option. If the last option does not confirm the connection, the new node becomes a CH and follows the steps explained in case 2. When a CM receives a “Welcome M” message, it will know which its cluster is. It will send “Keepalive M” messages periodically to the CH in order to indicate it is alive. If the CH does not receive a “Keepalive M” message from a CM for a dead time, it will erase that entry from the database. “Keepalive M” messages contain the *nodeID* of the sender, its λ and its δ parameters. Steps followed when there is a new CM in this third case are shown in Figure 3.38.

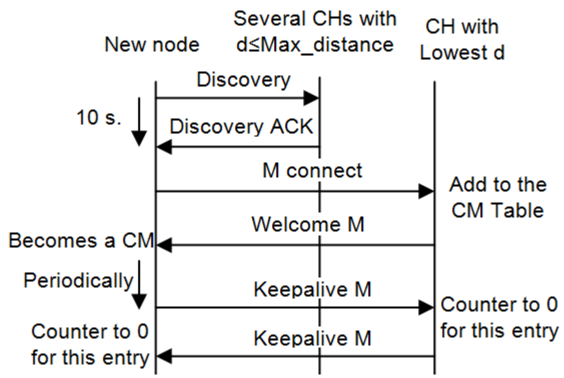

Figure 3.38. Protocol operation for a new CM.

When there is a new CM in a cluster, it has to establish connections with CMs from other clusters. They must be the same type of CM, and then, the distance between them has to be lower than or equal to than Max_distance (remember that it could be changed by the Received Signal Strength Indication (RSSI) value or by the Signal Noise Ratio (SNR) value or by any other parameter that can be used to know if the neighbor is reachable). First, it has to send a “CM request” message to the CH of its cluster. This message has the requester’s CM type and its position.

When the CH receives this message, it changes the *nodeID* by the *clusterID* in the message and forwards it. It is sent only to neighboring CHs because they are the only ones that will meet the Max_distance requisites. When a CH of another cluster receives that message, it sends the message to the appropriate CM (based on the type of CM). CMs that have not reached the Max_con value will send a "CM connect" message to the new CM. If it receives more than one "CM connect" messages from the same cluster, first it will choose the closest one, and in case of a draw, the one with highest λ parameter. Then, it will add these neighbors in its CM-CM table and will send them a "Welcome CM" message. They will add this entry to its CM-CM table. Finally, both will send "keepalive MM" messages periodically to indicate that they are still alive. If any one of them does not receive a "keepalive MM" message for a dead time, it will erase this entry from its database, so it will send a new "CM request" for this cluster. "Keepalive MM" contains the clusterID of the sender and its λ parameter. If the CM does not find any CM of the same type from any neighboring cluster, it will be alone until it receives a "CM request" message from the same type of node. Steps explained are shown in Figure 3.39. The whole procedure explained for a CH and a CM is shown in the flowchart of the Figure 3.40.

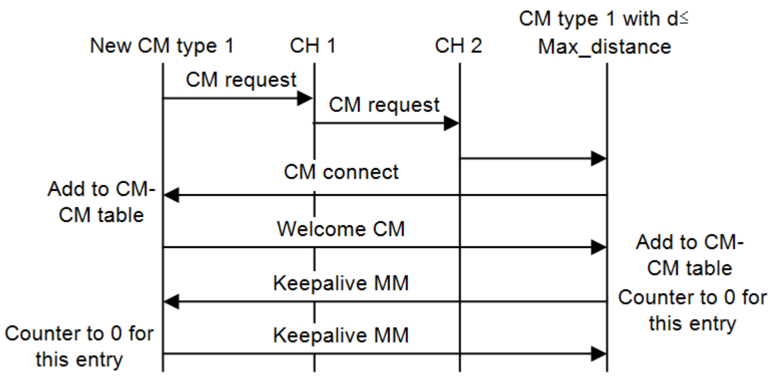

Figure 3.39. Protocol operation to establish a connection with a CM of another cluster.

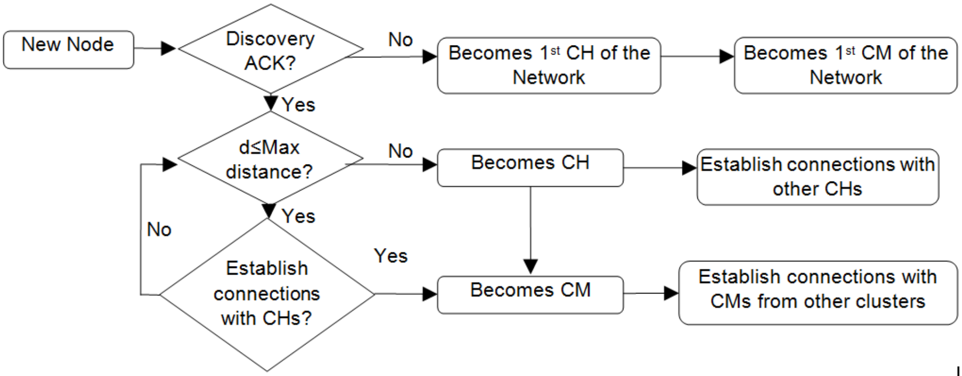

Figure 3.40. Flowchart of the architecture operation.

When a CM leaves the cluster (because of its mobility, or due to failure or other issues), the CH of its cluster and its neighbor CMs will not receive any "keepalive" message from it. After a dead time, the CH will erase this entry from its CM table and its neighbor CMs from other clusters will erase this entry from their CM-CM table. If the CM from the other cluster does not have any other neighbor for this cluster, it will send a "CM request" message to that cluster. Figure 3.41 shows all steps explained.

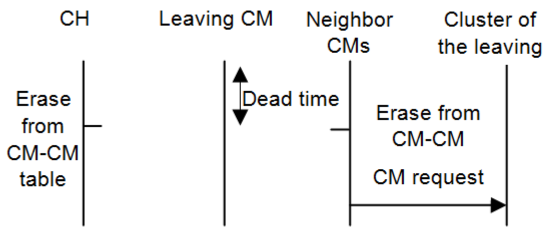

Figure 3.41. Protocol operation when a CM leaves the network.

Because the CH receives the δ parameter from all CMs in its cluster, it knows the best CM to promote in case of a failure or disconnection. The CM with highest δ parameter between the nearest ones is called "backup CH". The CH sends the CM table and the CH table though "backup" messages to the backup CH. The first message has all the tables; the next ones will only be updates. The CH sends "keepalive H" messages periodically to its neighbor CHs and to the backup CH. If the CH fails down, the neighboring CHs and the backup CH will know it due to the absence of keepalive messages. If the backup CH does not receive a keepalive message from the CH for a dead time, it will become the CH its cluster. The neighbor CHs of the failed CH will erase that entry from their CH table. That update will be propagated through the CH network using the OLSR routing protocol (although it can be changed by other routing protocol). Because the backup CH has both CM and CH tables of the failed CH, it will become CH and will send a "H replace" message to all CMs and CHs in the table to indicate they have to replace the failed CH by the new one, so it establishes, for its first time, a connection with the CMs in its cluster. Figure 3.42 shows the described procedure.

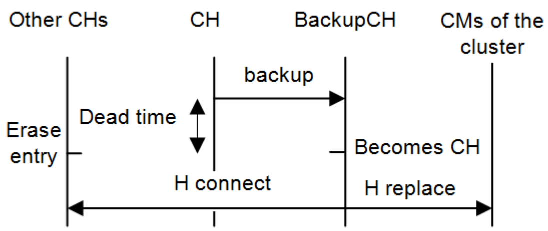

Figure 3.42. Protocol operation when a CH leaves the network.

3.4.5 Protocol Messages

In order to achieve the proper operation of the architecture, 14 messages have been designed and developed. We have used 4 Bytes for *clusterID*, *nodeID*, l and d parameters and the node position,

and 2 bytes for the type of message and the CM type. All are fixed size messages except the backup message that depends on the number of neighbors (although it is sent using incremental updates, there could be several new neighbors) and it is only sent when changes take place. Bandwidth cost in bytes for each message is shown in Figure 3.43 (1 = "Discovery", 2 = "Discovery ACK", 3 = "H connect", 4 = "Welcome H", 5 = "Keepalive H", 6 = "M connect", 7 = "Welcome M", 8 = "Keepalive M", 9 = "CM request", 10 = "CM connect", 11 = "Welcome CM", 12 = "Keepalive MM", 13 = "backup CH", 14 = "H replace"). The messages' size is based on MAC layer in 802.11 and TCP/IP headers. The sum of these headers is 70 bytes.

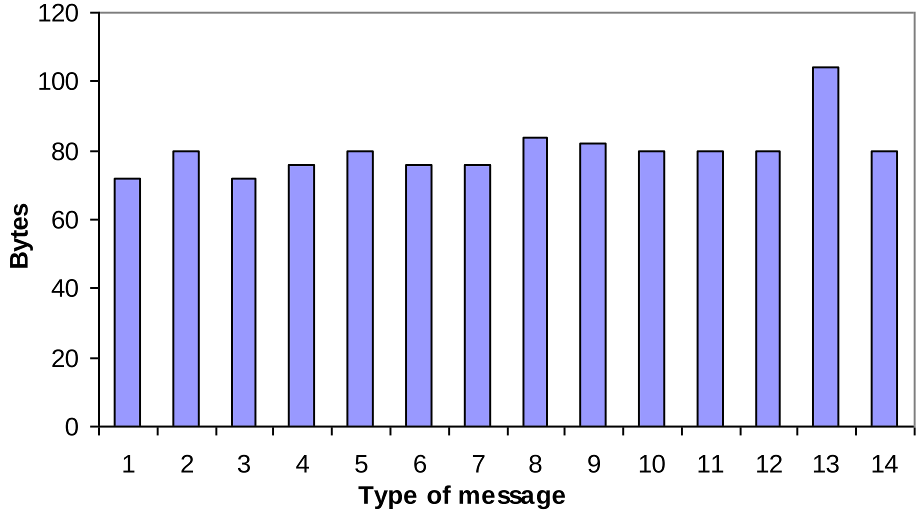

Figure 3.43. Bandwidth cost of the messages (backup CH message has 2 entries).

3.4.6 Architecture measurements

3.4.6.1 Test bench

In order to measure our proposal, an application software has been developed, using Java programming, to run and test the designed protocol and the architecture performance. We programmed CH and CM functionalities. The application allows configuring some parameters such as the maximum number of connections, maximum distance, type of node, node position, keepalive time and so on). The application calculates λ and δ parameters internally. The MAC protocol used to measure the control messages was CSMA/CA in the frequency of 2.4 GHz. There was one laptop with a high gain antenna configured in a monitor mode that captured all data of the test bench using a sniffer application.

The test bench is formed by 16 Multisensors [107] in an open air environment. Because we needed a multisensory with high computation capacities, we used the Linksys WRT54GL router, from Cisco Systems inc., as the core controller. It is an embedded system that that allows us to connect several physical sensors in its serial interfaces. The Linksys WRT54G version 4.0 has a 200 Mhz processor, 4 Mbytes of flash memory, 16 Mbytes of RAM at 100 MHz clock rate and 256 Bytes prefetch cache. The Wireless interface accomplishes IEEE 802.11g at 54Mbps and

IEEE 802.11b at 11Mbps standards. Its transmitting power is 18 dBm. It allows us large distances.

The position of the multisensor is shown in table 3.6. X and Y values are in meters. It is one of the most representative test bench performed because it allows showing several operation procedures as they will be seen later. The maximum number of connections has been fixed to a value of 4. The maximum distance to establish a connection with a CM of the same type was 100 meters. The keepalive time chosen was 30 seconds (it avoids too much energy consumption because of the number of messages sent and, if larger times are configured, nodes will discover the failures too late) and the dead time chosen was 60 seconds (twice the keepalive time). Sixty seconds seems to be enough time to know the node has failed. There are 4 types of nodes for each group. In order to make the test bench more understandable, CH nodes were CMs of the same type, but it does not affect to the system because they have to establish connections as CH and as CMs (the last ones are only established if the distance is lower than 100 meters, so in this case there will be none for type a). Then, the nodes are started in a sequential order every 10 seconds. This order is shown in table 3.6.

Table 3.6. Nodes' position

Node number	Type of node	X	Y
1	a	0	0
2	b	50	0
3	a	100	150
4	b	50	100
5	c	25	50
6	c	100	75
7	a	200	25
8	b	150	100
9	a	200	175
10	c	150	50
11	d	125	125
12	b	175	175
13	c	200	75
14	d	125	25
15	d	50	25
16	d	150	200

Table 3.7 shows the connections that have been obtained for every node when the network has converged.

Table 3.7. Neighbor connections.

Node number	Role	Connections with CH	Connections with CM
1	CH	3, 7, 9	2, 5, 15
2	CM	1	4
3	CH	1, 7, 9	4, 6, 11
4	CM	3	2, 8
5	CM	1	6
6	CM	3	5, 10
7	CH	1, 3, 9	8, 10, 14
8	CM	7	4, 12
9	CH	1, 3, 7	12, 13, 16
10	CM	7	6, 13
11	CM	3	14, 16
12	CM	9	8
13	CM	9	10
14	CM	7	11, 15
15	CM	1	14
16	CM	9	11

The topology obtained for the test bench and the physical position of the surrogates are shown in Figure 3.44. CHs have connections with other CHs (lines formed by black points). CMs have a connection with the CH of its cluster (lines formed by red points) and with the selected CMs of the other clusters (solid lines in different tonalities of grey and in black). CMs communicate with each other using a multi-hop architecture. We have implemented the positions using only 2 dimensions, but our application supports 3 dimensions.

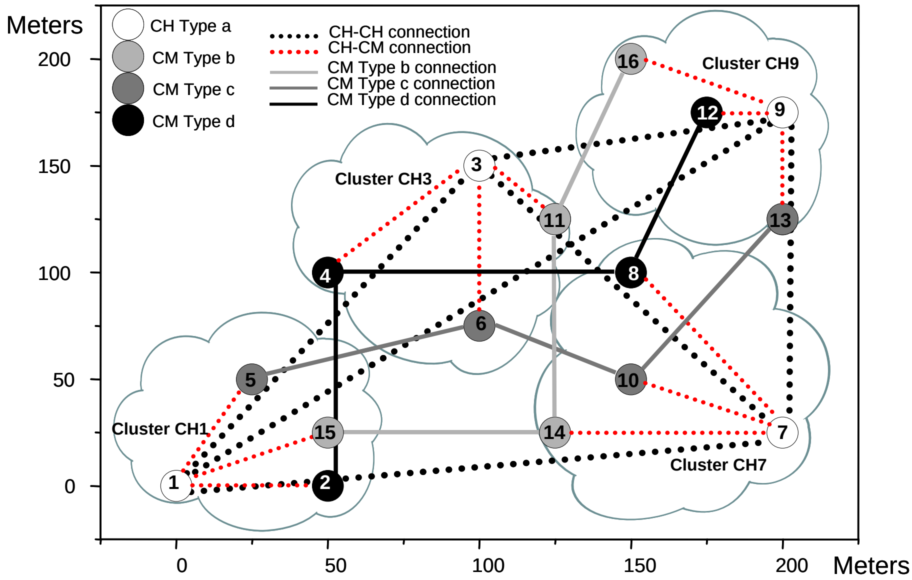

Figure 3.44. Nodes distribution and the connections established.

3.4.6.2 Network measurements

In order to check the performance of the developed architecture, the behavior of the nodes in the initialization phase has been measured for 10 minutes. It allows us to see how the network performs when the nodes join in. It gives us the amount of control traffic introduced by our architecture. The following graphs are the most representative measurements obtained over multiple experiments. We have performed several times the same topology obtaining very close results (we think that their difference is given because of the operative system response or due to electronic issues).

It can be observed that when all these nodes started sequentially, only 4 groups were created and the CHs were nodes 1, 3, 7 and 9. The number of broadcasts sent by the nodes when the network is setting up is shown in Figure 3.45. There are peaks of broadcast due to new joining nodes in the first 160 seconds, but the highest peaks had 8 broadcasts. There are no more than 2 broadcasts per second when the network is stabilized. So it demonstrates that there is low bandwidth consumption and little energy is wasted because there are few broadcasts.

Figure 3.46 shows the number of bytes/s in the network. At the beginning there are many peaks because of keepalive messages and joining nodes in the initial process. When the network has converged, there are peaks approximately every 60 seconds because of keepalive messages, but the number of bytes/s is very similar. It demonstrates that little additional traffic will be introduced when the topology changes. On the other hand, there is a mean value of 1482.5 Bytes per second, with a maximum value of 9823 bytes per second and a minimum value of zero.

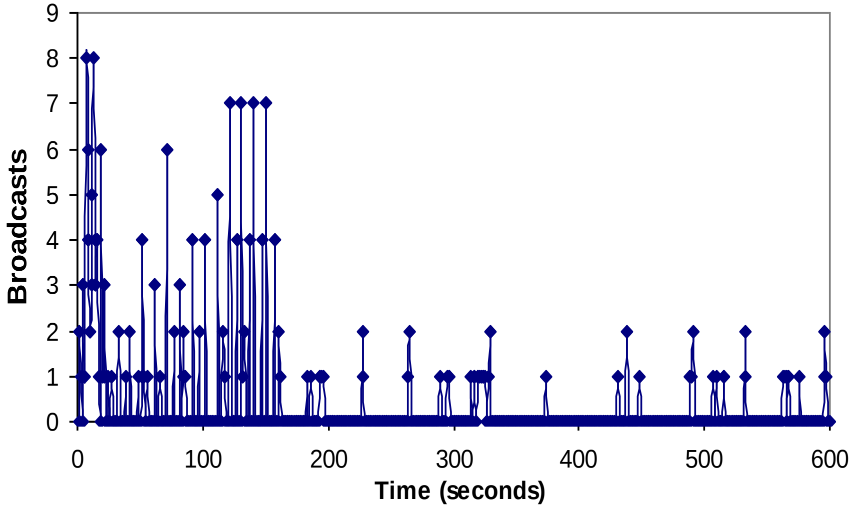

Figure 3.45. Number of Broadcasts per second.

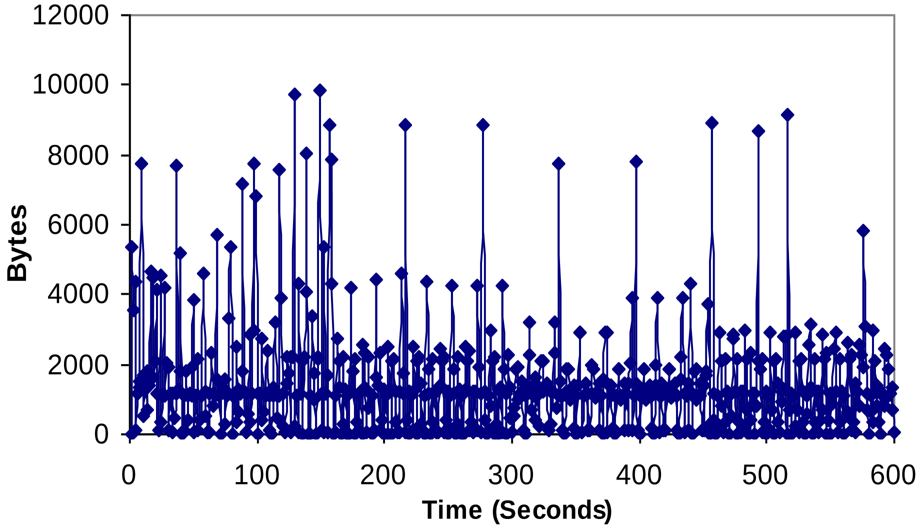

Figure 3.46. Number of Bytes per second.

Taking into account that the test bench was performed in an IEEE 802.11g Wireless LAN with 54 Mbps (16 nodes in 200 meters x 200 meters), we can state that, when our protocol is running, the limitation of number of devices in the network will be given by the overheads and timing constraints from all other network layers, not by our protocol.

It can be seen that there are more packets per second in the architecture when there are more clusters in the network (between 140 and 160 seconds can be seen in Figure 3.47). It is because new CMs request neighbor CMs of other clusters. Once the network has converged, there are not so much variations. The mean value has been 17.41 packets per second. We obtained a maximum value of 101 packets per second.

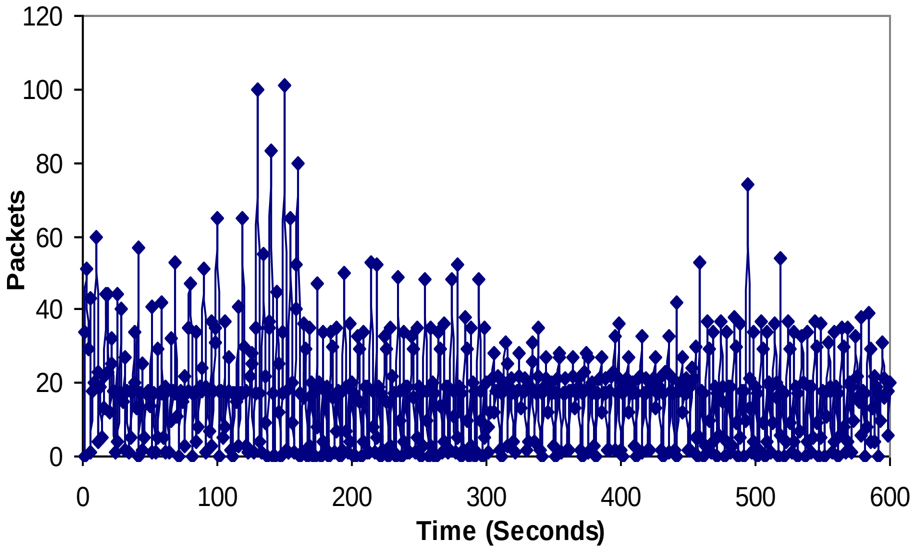

Figure 3.47. Number of packets per second.

3.4.6.3 Fault tolerant procedure

This subsection shows the procedure and the connections obtained after a node failure. It is divided into 3 scenarios to show different cases. The test bench shown in Figure 3.44 has been used as a starting point. The multisensors have had the same initial parameters as in the previous case. All multisensors have had the

same starting energy for simplicity and they have appeared in the scenario in the same manner as in the previous case. The traffic obtained in the whole process is not too relevant because there are very few messages transmitted through the network and they cannot be distinguished between the
traffic measured.

In the first scenario node 3 fails down one second after it sends its update message. Then, we sniffed the open air during 140 seconds (60 seconds was the configured dead time interval). This time makes us certain that node 4 had noticed that node 3 had failed down. It also gives us enough time to compare the update messages with the ones sent 120 seconds later.

Figures 3.48, 3.49 and 3.50 show the measurements gathered. We observe that there are more broadcasts/s, bytes/s and packets/s sent to the network between seconds 55 and 65 than between 115 and 125. We can see in Figure 3.48 that there was 18 broadcasts/s in the first range versus 15 broadcasts/s in the second range. Figure 3.49 shows that the first range has 3196 bytes/s sent to the network, while the second range has 3054 bytes/s. In Figure 3.50, 34 packets were sent in the first range while 32 packets were sent in the second range. The measurements taken let us know that, although there were more broadcasts/s, bytes/s and packets/s, the rise in the first range has been moderated, so when a node fails down, there is very low impact in the network.

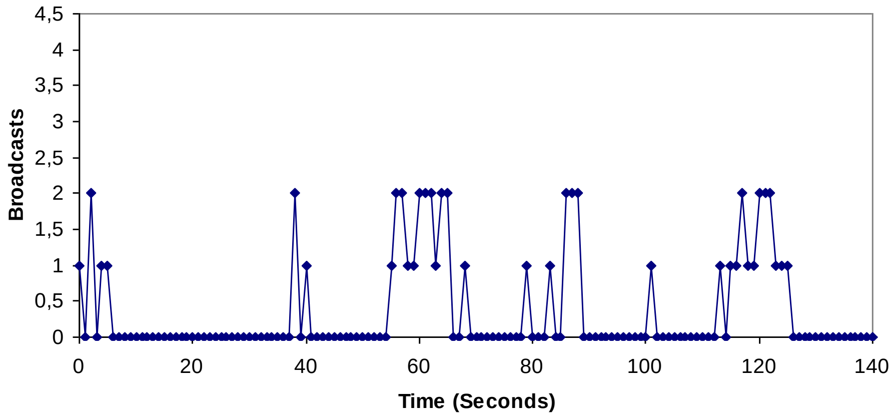

Figure 3.48. Number of Broadcast per second when node 3 fails down.

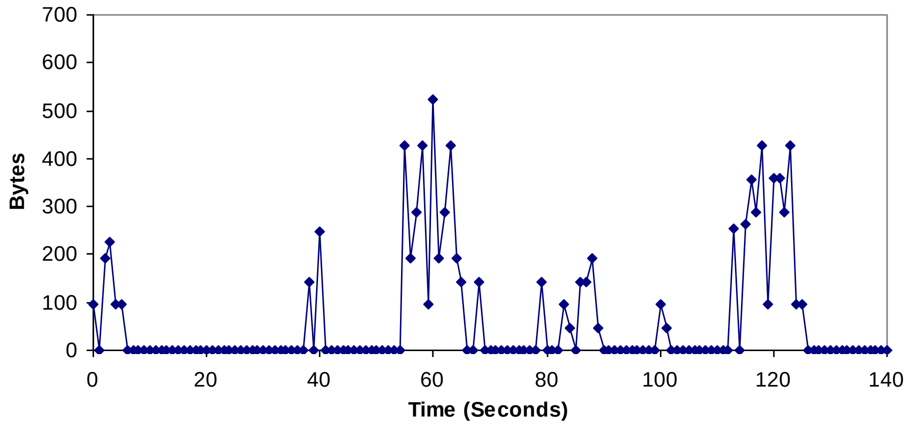

Figure 3.49. Number of Bytes per second when node 3 fails down.

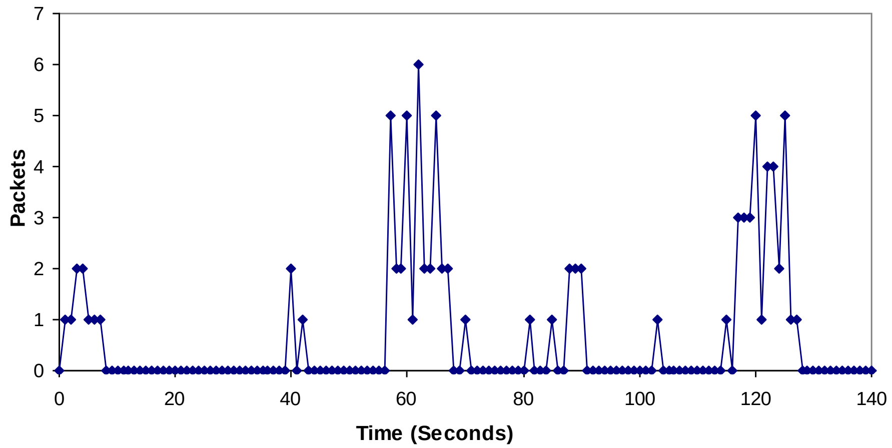

Figure 3.50. Number of Packets per second when node 3 fails down.

As all nodes have had the same initial energy, taking into account equation 1, it was deduced that node 4 would be the new CH because it had lower *nodeID*. Figure 3.51 shows the topology when the network has converged. Because node 3 only had connections as a CH, but not as a CM, when the network has converged, there is just one connection less, so it keeps the same stability.

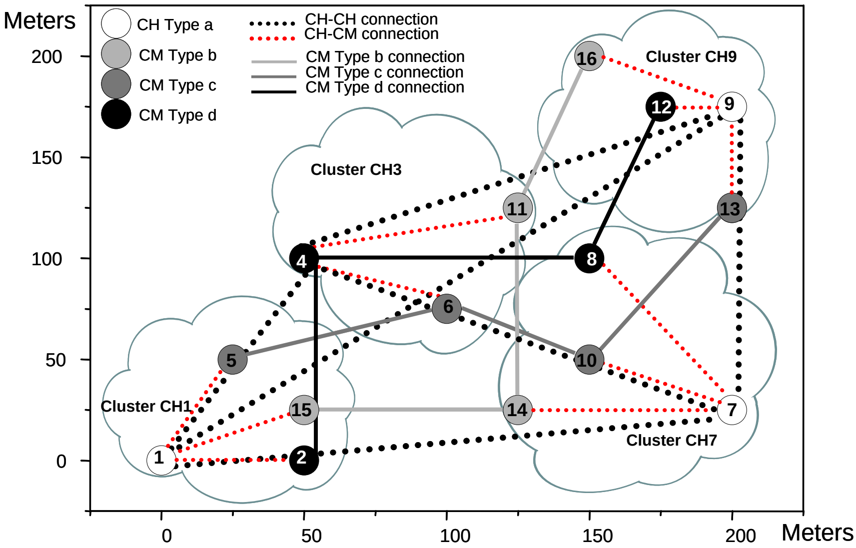

Figure 3.51. New topology when node 3 fails down.

In the second scenario node 15 fails down one second after it sends its update message. Then, we sniffed the open air during 140 seconds (60 seconds was the configured dead time interval). This time makes us certain that node 14 had noticed that node 15 had failed down. It also gives us enough time to compare the update messages with the ones sent 120 seconds later.

Figures 3.52, 3.53 and 3.54 show the measurements obtained when node 15 fails down. There is the same number of broadcasts/s, bytes/s and packets/s sent to the network between seconds 55 and 65 than between 115 and 125. In both ranges, we measured 15 broadcasts/s, 2958 bytes/s and 30 packets/s respectively. The measurements taken show that a CM failure does not have any impact in the network.

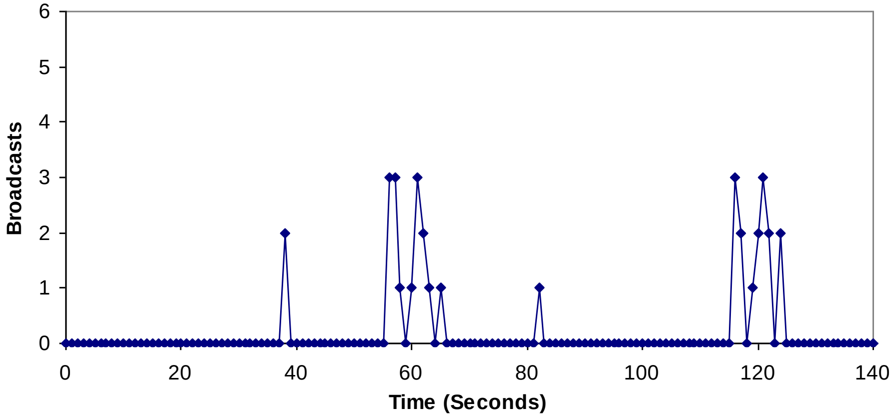

Figure 3.52. Number of Broadcast per second when node 15 fails down.

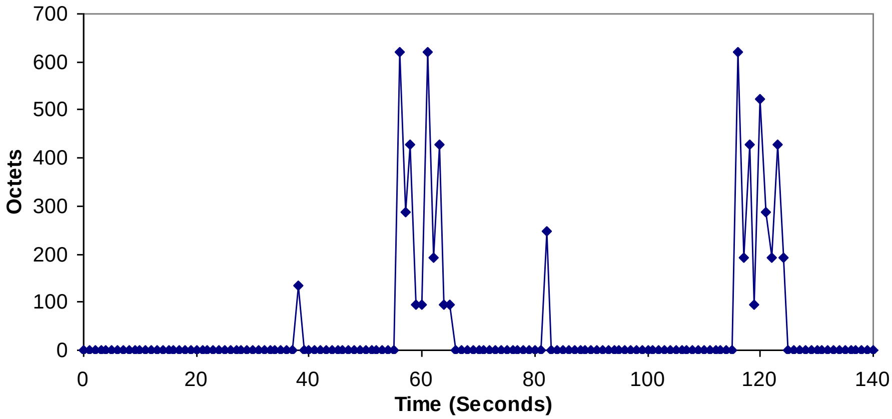

Figure 3.53. Number of Bytes per second when node 15 fails down.

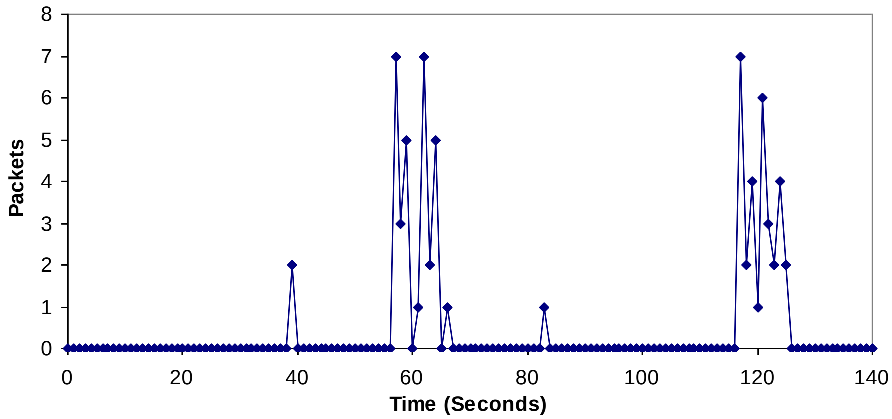

Figure 3.54. Number of Packets per second when node 15 fails down.

Figure 3.55 shows the network when it has converged. Now, we observed that there were two connections less: the connection between node 14 and node 15 and the connection between node 1 and node 15. It can be seen that there was no other type b node in the cluster CH1 to replace it and there are no other type b nodes close to it to have connections.

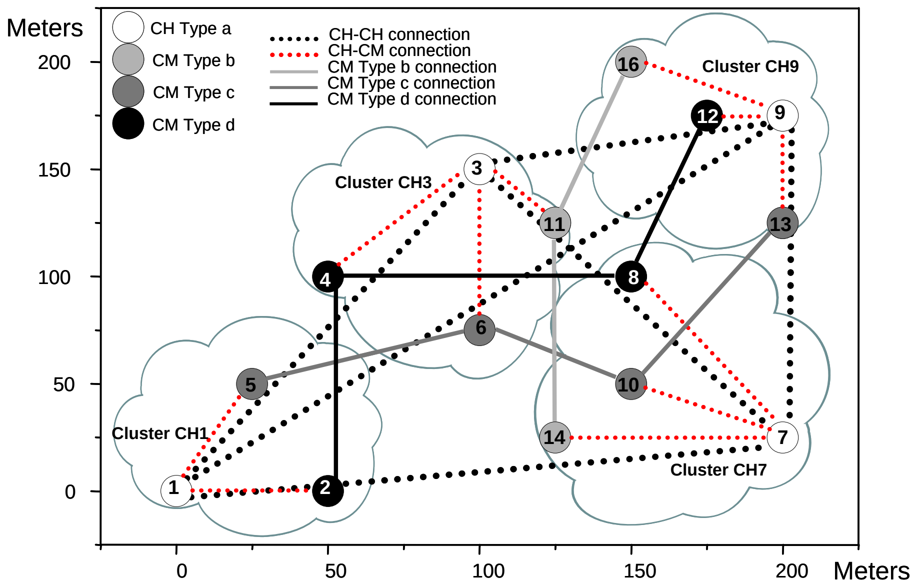

Figure 3.55. New topology when node 15 fails down.

In the third scenario node 6 fails down. After a second, we introduced node 17 in position (100,125), which is a CM of type c in order to provide a backup for cluster CH3. When we saw the connections in the network (see Figure 3.56), we discovered that the new node have node 10 and node 13 with distances lower than 100 meters, but node 5 was at a longer distance (it was not within the coverage area). In this case we don't provide broadcasts/s, bytes/s and packets/s measurements because they were similar to the one obtained when node 15 fails down.

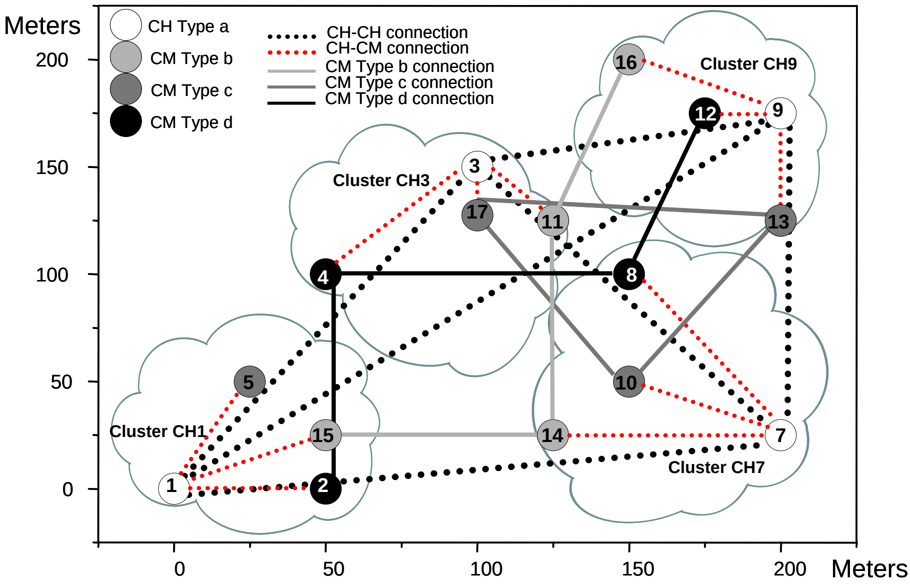

Figure 3.56. New topology when node 6 fails down.

3.4.6.4 Bandwidth, jitter, delay, lost packets and number of packets with errors

We took as a starting point the topology shown in Figure 3.44 but this time it was performed in a hard indoor environment with many Wifi networks working in parallel. In order to take these measurements, Wireshark network protocol analyzer was used [108]. First, we sent variable bitrate streams during 4 minutes from node 5 to node 13. The path followed was: node 5 – node 6 – node 10 – node 13. The bandwidth consumed is shown in Figure 3.57. It varied from 0 to 1551.26 Kbps. The mean value was 750.81 Kbps.

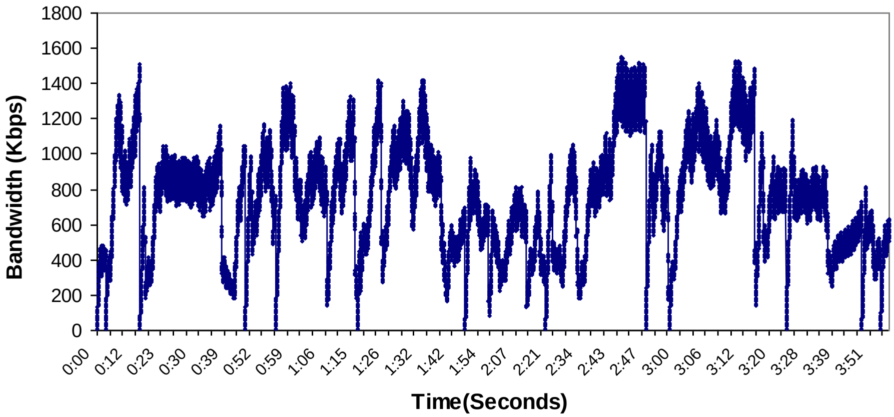

Figure 3.57. Bandwidth consumed during the test.

The measured delay is shown in Figure 3.58. We obtained a mean value of 19.16 milliseconds (which is a real time values because it is lower than 50 milliseconds). The delay varied from 0 to 1826.41 milliseconds.

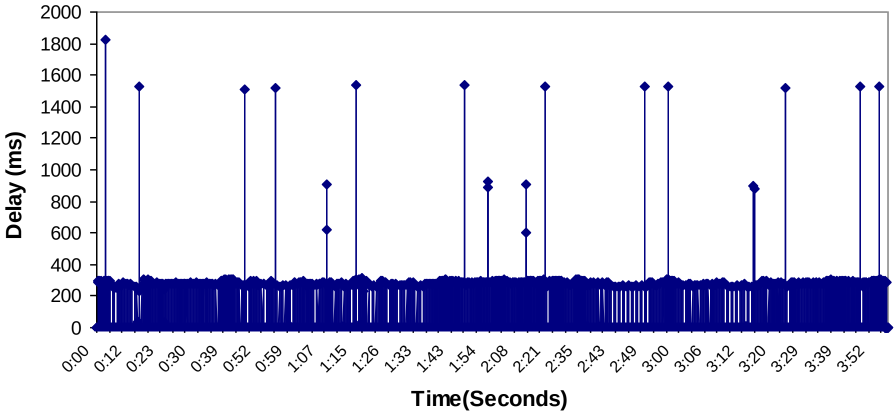

Figure 3.58. Delay measured during the test.

Then, we measured the jitter of the packets during the test. It is shown in Figure 3.59. We obtained a mean value of 31.11 milliseconds. The maximum value was 171.14 milliseconds and the minimum value was 0 milliseconds.

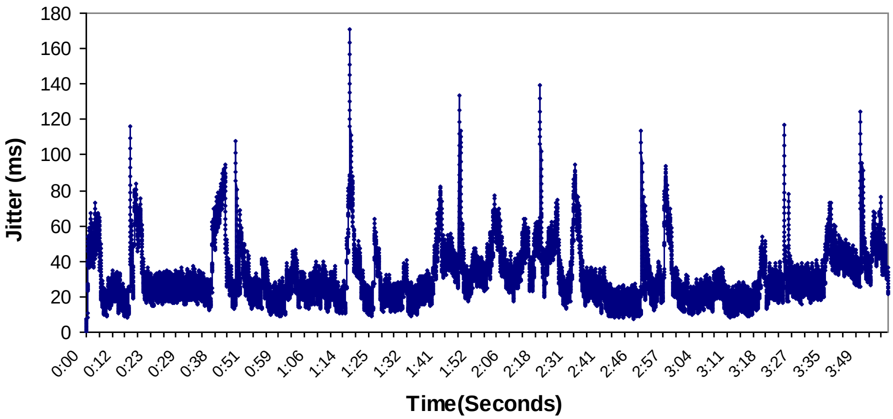

Figure 3.59. Jitter measured during the test.

During this test, 15537 packets were sent, 1475 of them were lost (9.49% of dropped packets), and 183 of them had sequence errors. Those values are not so bad if we take into account that 8 IEEE 802.11b/g networks were detected in the place where we performed the test. Packet loss can be caused by one of the following: signal degradation over the air, oversaturated network links, corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers or normal routing routines.

3.4.7 Protocols Comparison

Cluster architectures have been proposed for many different purposes. All of them have the following benefits in common:

- Topology updates overhead reduction.
- The clustering structure is self-organized and adaptable.
- Fully distributed operation.
- New nodes do not have to be searched or initiated.
- Broken routes cloud is repaired locally without rediscovery.
- Reduction in the energy dissipation.
- Broadcasts are done only by the boundary nodes.

- The routing is source initiated.
- Lower memory overhead.
- Quite scalable.

There are many other characteristics that could be different. The cluster scheme can be applied in many different manners in order to achieve diverse benefits. This section compares our proposal with others in existence in order to show the benefits of our protocol. Only those architectures whose protocol has been available and accessible are included.

3.4.7.1 Architecture comparison

P. Krishna et al. presented a methodology for routing and topology information maintenance in mobile wireless network based on the existence of clusters in random graphs in [109]. They divided the graph into a number of overlapping clusters. There are no cluster heads in the proposal. It is an on- demand source routing. The performance of the routing protocol proposed by them is determined by the average cluster size. The effectiveness of their approach lies in the fact that any routing protocol can be directly applied to the network, replacing the nodes by clusters. They designed 9 messages, but their approach has to be implemented over another routing protocol. They proposed a standard distance vector routing protocol to apply their proposal. Taking into account that AODV has 7 messages. Their implementation needs 16 messages.

CBRP (cluster-based Routing protocol) was proposed in [87]. The protocol divides the nodes of the ad hoc network into a number of overlapping or disjoint 2-hop-diameter clusters in a distributed manner. CBRP uses IP Protocol for routing purposes and interoperability with fixed networks. It uses 6 messages plus the ARP messages, so it needs 8 messages to work properly. As a source routing protocol, there is an overhead bytes per packet.

In [110], the authors proposed a Cluster-Based Security Architecture for Ad Hoc Networks. They proposed a division of the network into clusters, with one special head node each, for a distributed public key infrastructure. These cluster head nodes execute administrative functions and hold shares of a network key used for certification.

KCLS protocol was proposed in [111]. The paper describes a location-service protocol based on the clustering architecture, which is able to balance the tradeoff between the communication overheads and the accuracy of location information. It has the capability of cluster-level self-route recovery against interlink failures. Taking into account that KCLS is based on the KCMBC protocol and on a Link State protocol, it needs 13 messages plus the protocol needed to acquire their position using GPS.

In [50] the authors proposed an adaptive clustering scheme for spatial reuse of the bandwidth (relying on a code division access scheme) for multimedia support in Mobile Wireless Networks. They use only one code within each cluster. The clusters are independently controlled and are dynamically reconfigured as nodes move. Bandwidth can be shared or reserved in a controlled fashion in each cluster.

LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol was proposed in [106]. It is a clustering-based protocol that utilizes randomized rotation of local cluster-heads minimizing the global energy usage by evenly distributing the energy load among the sensors in the network. LEACH uses localized coordination and incorporates data fusion into the routing protocol to reduce the amount of information that must be transmitted to the base station. It is completely distributed, requiring no control information from the base station, and the nodes do not require knowledge of the global network in order to operate.

Reference [112] presented the "Base Station Controlled Dynamic Clustering Protocol" (BCDCP). Their proposal utilizes a high-energy base station to set up clusters and routing paths, perform randomized rotation of cluster heads to avoid cluster head overload, and carry out other energy-intensive tasks. It distributes the energy dissipation evenly among all sensor nodes to improve network lifetime and average energy savings.

In [113], authors proposed CLACR (Core Location-Aided Cluster-Based Routing Protocol for Mobile Ad Hoc Networks). CLACR splits the network into square clusters. Cluster heads compute the desired route using Dijkstra algorithm, which reduces the number of nodes participating in routing, the routing traffic and the route setup time.

CBLARHM (Cluster Based Location-Aware Routing Protocol for Large Scale Heterogeneous MANET) was proposed in [114]. The system uses the geographical location information of mobile nodes, provided by global positioning systems (GPS), to confine the route searching space, for the specified destination node.

WCA (A Weighted Clustering Algorithm for Mobile Ad Hoc Networks) was presented in [57]. They proposed an on-demand, weight-based distributed clustering algorithm that takes into consideration the ideal degree, transmission power, mobility, and battery power of mobile nodes. The clustering algorithm tries to distribute the load as much as possible aimed to reduce the computation and communication costs. The algorithm is executed only when there is a demand, i.e., when a node is no longer able to attach itself to any of the existing cluster-heads.

In [115] CLTC was presented, a Cluster-Based Topology Control Framework for Ad Hoc Networks. CLTC uses a centralized algorithm within a cluster and between adjacent clusters to achieve strong connectivity. It utilizes a hybrid approach to control the topology using transmission power adjustment and yet achieves the scalability and adaptability of a distributed approach with localized information exchange between adjacent clusters. CLTC framework guarantees global k-connectivity as long as the original topology is k-connected.

Table 3.8 compares the described protocols. The number of messages of some protocols is provided by the explanation of their authors in the referenced paper. In Table 3.8, n/a means that it is not applicable and n/p means that it has not been provided by the authors. Some of them do not take into account messages such as the new nodes messages or messages to provide fault tolerance, or are based on other algorithms not described in the paper (although in some cases we have found and taken them into account). Many other proposals have been found in the literature, but they have not been included because of the lack of information in their publication to fill up the rows of the table 3.8, or because the authors just described the algorithm, not the protocol, or because they are light extensions of the proposals shown in the table.

Table 3.8. Cluster architectures comparison.

Architecture	Overlapping nodes	Uses other routing protocols	Number of messages	New node cluster's selection	Purpose	Node Fault Tolerance	Cluster Head election
P. Krishna et al. [109]	Yes	Just one at a time	16	Proximity	Routing	No	n/a
CBRP [87]	Yes	No	6	nodeID	Routing	No	Lowest node ID
Marc Bechler et al. [110]	No	Just one at a time	n/p	n/p	Security	Yes, but very weak	Trusted node, but not explained what happens if there are several trusted nodes.
KCLS [111]	No	No	13 + GPS protocol	Distance to the head node less than k hops	Location Service	Yes	Mobility threshold
Chunhung R. Lin [50]	No	Just one at a time	n/p	nodeID	Bandwidth allocation	Yes	Lowest node ID
LEACH [106]	No	No	n/p	received signal strength	Energy optimization	No	Random rotation

Note: n/a means not applicable, n/p means not provided.

Table 3.8 (Continued). Cluster architectures comparison.

Architecture	Overlapping nodes	Uses other routing protocols	Number of messages	New node cluster's selection	Purpose	Node Fault Tolerance	Cluster Head election
BCDCP [112]	No	No	n/p	Location	Energy optimizat ion	No	Random
CLACR [113]	No	No	11	Location	Routing	Just for location servers	Closest to the cluster center position
CBLARHM [114]	No	No	17+ GPS	Relative distance, velocity and time	Routing	No	Node-Weight heuristic
Automatic Decentralized Clustering [115]	No	Just one at a time	n/p	Degree difference and proximity	General purpose	No	Combined weight metric
CLTC [116]	No	Just one at a time	10+ GPS	Coverage area	Topology control	No	n/p
Our proposal	No	Yes and it could use many simultaneousl y.	14	Proximity + capacity parameter	Create parallel networks	Yes	Promotion Parameter

Note: n/a means not applicable, n/p means not provided.

Several features are highlighted in our proposal. First, it is the only one that is able to use several routing protocols in the same network, and it does not depend on a specific routing protocol so it could be adapted to the environment issues. It does not have too many messages compared with the other ones (despite the simplicity of the CBRP, some procedures are not explained and it does not provide fault tolerance). Second, it is the only one where a new node selects the cluster not only by the proximity or radio signal strength but also takes into account the available capacity of the neighbors (which depends on the available energy). Third, it is the only one that has been proposed to create parallel networks. Our proposal has been designed to provide fault tolerance and our design is described in detail giving all the messages needed to run properly.

There are several differences in the metric used to elect the Cluster Head. Some of them use the lowest *NodeID* or their position, while others use just a random system. Others are not explained or propose a weak system. Between the most complex metrics, we distinguish CBLARHM that uses a node-weight heuristic parameter, based on the ideal number of nodes in a cluster, the battery power, the average link stability and the average dependency probability, to elect the head cluster node, but this is very impractical because it is difficult to determine the value of some of these parameters. WCA uses a combined weight metric based on the ideal node degree, transmission power, mobility and the battery power of the nodes. It is a good idea and seems very useful, but they propose this metric for a cluster-based general purpose algorithm, may be these parameters are good for a specific case, but not for all cases, some parameters could be missing such as the node's position or the node's load. Our metric does not take mobility into account since the entire cluster could be moving and avoids continually selecting the motionless nodes. On the other hand, it does take into account the more stable node in the cluster and its energy, thus making the system very simple and practical.

3.4.7.2 Measurements comparison

First of all, we want to emphasize that all works found in the literature provide only measurements taken from simulations, neither from real deployments nor from controlled testbeds.

Table 3.9 gives the type of measurements provided by several papers in the literature. Some of them are focused on measuring parameters related with the cluster size and the number of clusters. Due to cluster-based networks are mostly used for energy saving, most of them simulate energy issues.

Table 3.9. Type of measurements provided by other authors.

Reference	Type of measurements	Purpose
[109]	It gives the simulations of the average cluster size and the number of clusters versus the degree of the nodes.	Parameters related with the cluster size and the number of clusters
[114] and [116]	They give the average number of cluster heads versus the number of nodes and the cluster size versus the number of nodes.	
[117]	It gives the average number of nodes per cluster versus the transmission range.	
[90]	It provides the energy consumption versus the distance to the nodes and the number of nodes, the number of alive nodes versus the round and the energy consumed versus the number of rounds.	Energy issues
[95]	It shows the energy consumption versus the number of L-sensors, average number of working nodes versus the number of sensors and the energy consumption versus the average desired coverage degree.	
[106]	It gives the number of nodes alive versus the time, the energy dissipation versus the percentage of nodes that are cluster heads, and the energy dissipation versus the network diameter.	
[111]	It compares several cluster-based protocols versus the number of rounds, number of messages received versus the energy dissipation and energy consumed and the number of nodes alive as a function of network area.	
[118]	It shows the measurements of the number of alive nodes versus the time, the energy dissipation versus the time and the energy dissipation in the setup phase.	
[119]	It shows the percentage of energy consumed versus the cluster radius, the average cluster head residual energy versus the cluster radius and the cluster energy dissipated versus the number of nodes.	

Taking into account that the more message transmissions, the more energy dissipation, only CBRP [87] and CLACR [114] (and may be CLTC [116], depending on the number of messages transmitted using GPS) will consume less energy than our proposal, all the others will have higher energy consumption than ours.

Taking into account the measurements provided, we will compare our measurements with some measurements provided in other authors in their works. We are not going to implement the same test bench; we will just take their measurements and compare them with our measurements in some particular cases.

Studying the paper presented by M. Bechler et al. in [110] and, taking into account the same keepalive interval (30 seconds), we obtained an average value of 17.41 packets per second for a topology of 3 hops, while they obtained higher values (between 50 and 100 packets per second) in a random topology with 15 nodes. In terms of overhead (packets per second), it gives us more than 65.2 % of improvement compared the best case of their protocol.

Hollerung, T. D. presented in [119] several graphs that show the packet delivery ratio versus the number of nodes in the cluster network. Close to 1 packet delivery ratio (0.99 packet delivery ratio) was shown for 25 nodes, while we can see that it agrees our measurements. Once our network has converged (after the setup phase), we obtain 15.9 packets/s for 16 nodes. It gives us almost the same packet delivery ratio. The worst case has been presented in [116] by Shen C.-C. et al. because they measured 2.5 messages per node for 100 nodes.

In a previous sub-section we have obtained a mean value of 19.16 milliseconds when there were 3 hops between the source and the destination. In [105] we can see that the average delay for 3 hops was higher than 500 milliseconds in the best case (and lower than 1500 milliseconds in the worst case).

3.5 Conclusion

We can split the conclusion in three parts according to the sections included in this chapter.

On one hand, we have presented a protocol to join multimedia networks that are sharing the same type of resources or content. It is based on two layers: Organization and distribution layer. First one joins all multimedia networks, organizes nodes in zones and helps to establish connections between Dnodes. Second one allows send searches and data transfers between networks. Once the connections are established, content or resource searches and data transfer could be done without using organization layer nodes. We have defined several parameters to know the best node to promote to higher layers or to connect with. The recovery algorithm when any type of node leaves the architecture or fails down is also described. We have shown that messages with more bandwidth are the backup message and the one which sends the topological database, so, they are maintained by incremental updates and we have designed taking into account that only first message will have all information. Measurements taken, for 2 and 3 multimedia networks interconnection, show the time needed when a Dnode fails. The protocol does not consume so much bandwidth.

On the other hand, 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 remoteness parameter 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 not the network layer protocol. We have shown the performance of the network and how surrogates perform in different cases. We have 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. Measurements show that our proposal scales very well.

Moreover, we have shown the development of an architecture that creates clusters and establishes connections between sensors of the same type by building different sensor networks. Cluster heads manage the network since they have connections with other cluster heads and these connections allow connecting cluster members from different clusters. Cluster members of the same type form a specialized network. Although there are several proposals of cluster-based systems in existence, the novelty of our proposal is that it could be used to build different networks with different routing protocols, while other cluster-based networks can run just one routing protocol and can build only one type of network. One of the main goals is that if all cluster heads switch off at the same time, the system is able to continue working, although there will not be new connections between clusters through CHs. We have presented the description of the protocol developed and the flow of the messages. The performance of the network and how nodes perform in different execution cases have been shown. It has been demonstrated that the architecture requires low bandwidth to run and work properly. We are currently implementing the protocol in an embedded sensor. Comparing our proposal with others, it can be seen that the detailed description of our protocol allows its implementation easily. Other protocols compared in this section do not take into account the discovery algorithm, and/or the node starting procedure, and/or the fault tolerance and/or even some parts of their proposal are not described in detail, so more messages that are not included in their description will be needed.

The three sections included in this chapter have been published in international journals or well known conferences. They have been listed in the section “publications derived from the PhD”, in the conclusion chapter.

Chapter 4. Architecture and Protocol for Multimedia Wireless Ad Hoc Networks

4.1 Introduction

In this chapter we propose a Multimedia Wireless Ad Hoc Networks Cluster Architecture, named MWAHCA, and protocol. The proposed model and the developed protocol are the main contribution on this dissertation.

The first part of the chapter presents the architecture characteristics of a wireless ad hoc network required to optimize the multimedia delivery to improve both objective (QoS) and subjective (QoE) measures. Then, we present and detail the features and functions of the components of the MWAHCA architecture. It is specially designed to meet the requirements of wireless ad hoc networks. It provides the required mechanisms to be able to adapt the constantly changing network characteristics in real network environments. In the network, the appropriate level of QoS and QoE on multimedia transmission is guaranteed while the communication remains active. Resource reservation techniques are added in order to improve the service performance. The concept of Multimedia Init Profile (MIP) is introduced to identify the characteristics of a multimedia flow and it becomes an essential information container used to build the topology structure.

4.2 Multimedia-Oriented Architecture and Protocol for Wireless Ad Hoc Networks

Despite of Mobile Ad-hoc Networks (MANETs) benefits, they present several problems due to the low bandwidth, the shortage of resources, battery limit, etc. The topology of these networks change continually because mobile nodes can join or leave the network at will. In order to provide a real-time communication in MANETs, Quality of Service (QoS) must be included in such networks [120]. Even the size of the ad hoc network beyond a certain distance becomes an issue, because of the increased computational load and difficulties in propagating network updates within given time bounds.

Nowadays, the features of smart devices are very similar to the regular computer features. Last generation of tablet PCs and Smartphones can include advanced models of CPUs, with multiple cores, such as A6, Exinus, Tegra 3. In addition, the amount of RAM can be around 1 GB and the storage capacities can reach 64 Gb. Moreover, they include multiple wireless connections such as Bluetooth, Wi-Fi, 3G, 4G, etc.

According to Cisco [121] the global mobile traffic grew 70% in 2012. The number of mobile-connected devices will exceed the world's population in 2013. This will involve that the mobile data and fixed traffic will have the same growth rate. However, the mobile data growth rate is likely to remain higher than the fixed growth rate over the next decade. The number of devices connected to IP networks will be nearly three times higher than the global population in 2016 [122].

A consulting report to the 20 most important mobile operators in the world in 2012 [123] highlighted that smart phone sales increased over 40% of all handset sales, fuelled by low cost Android devices that are rapidly eating away feature phone market share.

Maybe, one of the most investigated issues regarding to the mobile hardware features is the energy consumption in these types of devices [124] [125]. There are several studies providing power saving and energy optimization techniques to improve their lifetime [3].

Regarding to the transmission content, multimedia is increasing every day. There are a lot of media platforms and protocols used in different fields, from entertainment to training in business environment [126]. These platforms have been developed to be used at home and business environments. Some examples of these platforms are YouTube or Shazam and VoIP and Video over IP applications such as Skype and IPTV.

Multimedia has very strict requirements because users always require an acceptable multimedia service, so applications must take into account the specific QoS requirements. The main goal of QoS is to achieve a more deterministic network behaviour [127]. If we want to provide better quality, we have to improve the use of the available network resources [128]. Some of the most important parameters to determine QoS include response time, delay, jitter, data rate, required bandwidth, loss rate and error rate. In order to provide an acceptable QoS in the network, we should define the values of QoS metrics in order to help to establish the necessary requirements [129]. These requirements are different if it is a real-time service or an on demand service.

A way to improve multimedia distribution in ad hoc networks is choosing the appropriate routing protocol. In the related literature we can find several types of routing protocols: pro-actives, reactives, flow-oriented, Hybrid (both pro-active and reactive) and hierarchical routing protocols. Either of these protocols could be used in our proposal to deliver multimedia service inside an ad hoc network.

We propose a new multimedia-oriented application layer protocol that takes into account the multimedia services offered by the nodes in the wireless ad hoc network to select the best multimedia service provider node inside the network to provide the best QoE and QoS to the nodes participating in the ad hoc network.

4.2.1 Multimedia-oriented architecture description

Our multimedia-oriented architecture and protocol is focused on providing the highest multimedia service quality in an ad hoc network. We will assume that lower layers of the communication stack are able to run properly, so we will not take care of the physical layer technology used in the ad hoc network (we will assume that the physical technology allows higher bandwidths than 1 Mbps), the medium access control protocol used to allow wireless links in a multiple access medium, and the network layer protocol, that allow having a multi hop network while allows delivering IP services over the ad hoc network. We will use the protocol stack shown in figure 4.1. It shows a regular ad hoc protocol layer model, but there is a clear transport layer which allows the coexistence of different multimedia protocols with different requirements (connection oriented, connectionless oriented, etc.) and there is a protocol in the application layer which interacts with the regular multimedia protocols and allows making service access point calls.

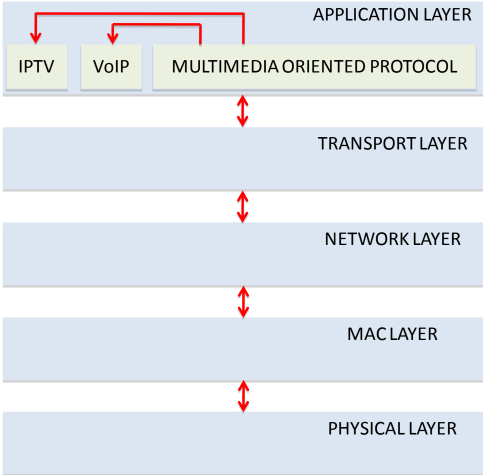

Figure 4.1. Multimedia-oriented protocol stack

In our ad hoc network any node could be the gateway of the network and provide IPTV, VoIP or multimedia services from Internet. Thus nodes can appear in the network at will and even provide multimedia services at will, once they are in the ad hoc network. There is no base station or infrastructure in the wireless network to deliver multimedia services, so any communication is performed in a wireless ad hoc manner.

In order to propose a protocol for a real environment, we will suppose that there is a heterogeneous network, where devices have different capabilities. Moreover, it will be asymmetric, because devices may have different bandwidth connection and offer different multimedia services (and thus different throughput needs). The ad hoc network is able to self organize and adapt to any change or event. It allows the spontaneous formation of the network. An explanation of how a network can be formed spontaneously is shown by us at [130]. Moreover, this protocol can be secured by using trust techniques when users join the network [131] and when the information is routed through the network [8].

Once a device joins the ad hoc network, it is able to share any multimedia service such as VoIP, IPTV, radio streaming, etc. or select a service offered to the network by other nodes. The service is announced using the regular application firm, e.g. Video LAN VLC, Real Media, etc., and the IP of the node offering this service. When a node detects the same service offered by other nodes, the requester node will always choose the one that provides better QoS and QoE parameters.

This change is made automatically by the application software installed in the mobile devices.

4.2.2 Analytical model.

The system can be modeled as follows. In our ad hoc network devices could join or leave the network at will or because energy constraints, so we will not take into account the energy constraints of the devices in our analytical model because any device may leave the network before its energy is consumed.

We assume that the ad hoc network has a limited lifetime. So we will divide the total time into individual periods of time, which are represented by $t \in T$ for t=0,1, …t_z. At time t_z, next to last node leaves the network (last node cannot form a network only by itself). Let's suppose that at time $t=t_0$ a device enters the radio coverage area of another device and, therefore, they become neighbors. Let R be the random variable that represents the time when a new node appears in the network. Responsiveness of the discovery process, $F_R(t)$, can be represented by the probability function of the random variable R, which is defined as is shown in equation 4.1. It is the probability of having an R value equal or lower than t.

$$F_R(t) = P\{R \leq t\} \qquad \forall t \geq t_0 \tag{4.1}$$

Now, let us know when a device in the network will be able to enjoy a multimedia service provided by a new device. In order to estimate it, figure 4.2 is provided. It shows the reference times used in our analytical model. First, a device V_g joins the network in t_g. It cannot enjoy any multimedia service until there is one announced in the system. At t_i a new device V_i joins the network. The device announces a multimedia service at t_1 after it has joined the network. Device V_g requests the announced multimedia service at t_2 and enjoys it. The new device announces the end of the service at t_3 and leaves the network at t_4 after it has joined the network, so this multimedia service is not offered in the network. Device V_g leaves the network at t_5 after it has joined the network. At t_z the network ends.

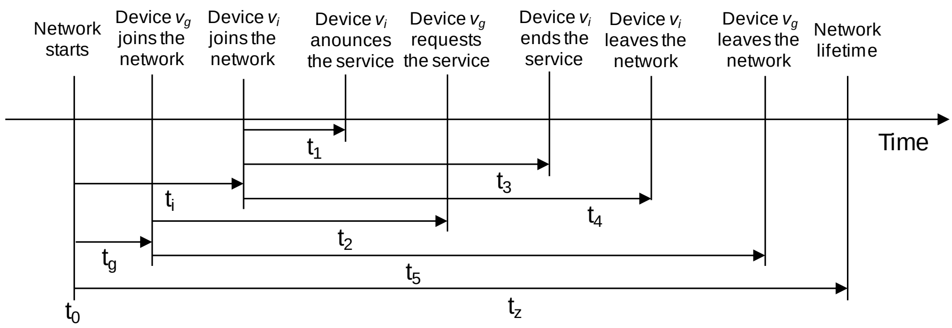

Figure 4.2. Reference times when a device v_g joins and leaves the ad hoc network

The service probability, $S(v_i)$, is defined as the probability that a device v_i has to hear any browsing requests and to announce its service during its appearance in the network. Using the information shown in figure 4.2, $S(v_i)$ is defined in equation 4.2.

$$S(v_i) = \Pr(t_2 + t_g > t_1 + t_i)\Pr(t_3 + t_i > t_2 + t_g) = \int_0^{t_z} \Pr(t_2 + t_g > t_1 + t_i)\Pr(t_3 + t_i > t_2 + t_g)dt \qquad (4.2)$$

The number of nodes in the ad hoc network is *n*. For simplicity, we will suppose that a new node announces a service, and the other nodes enjoy the service.

4.2.3 Protocol and algorithm operation

In order to join the proposed ad hoc network, the ad hoc network shared key must be known. It is a basic security system, but stronger mechanisms may be added easily [132]. When a new device joins the ad hoc network, it broadcasts a *discovery* message in order to know the devices under its coverage area. Devices replying the *discovery* message (with a *discovery ack* message) are candidates to be neighbours of the new node. Then, new node sends an *authentication* message to the selected neighbour devices. When neighbour devices receive this message, they will authenticate the new device according the shared key with an authentication ack message. If the device is sharing any type of multimedia service, it will broadcast a *multimedia service offer* message to the network, which will be forwarded to the whole network according to the implemented routing protocol. When it receives a request for this service, it will send the video or audio streaming accordingly. This service will be offered to the network devices till it stops the service or leaves the network. The exchanged messages are shown in figure 4.3.

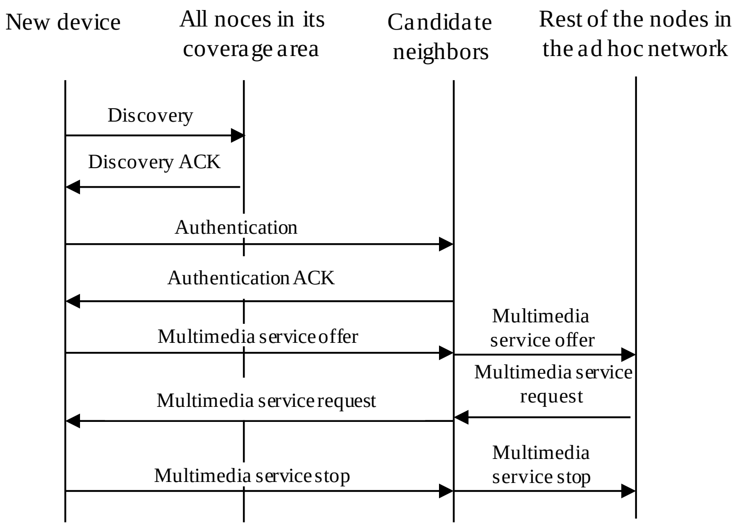

Figure 4.3. Messages exchanged when a node joins the network and offers a multimedia service

When a new node joins the network and wants to enjoy an offered multimedia service in the ad hoc network, after authenticating, it will receive a message from its neighbours with the services offered in that network (the same service may be offered by different nodes, but the application software will only display one offered service). The application of the mobile device will decide which node will be selected to serve the multimedia service. This selection will be firstly based on the number of hops from the requesting device to the multimedia service server, because we wish to keep less overloaded the network, and the secondly, in case of multiple paths, the selected one will be the one that has less delay, jitter and packet loss, and higher bandwidth. The equation used to take this decision was proposed by us for QoE estimation in [133]. This procedure is shown in figure 4.4.

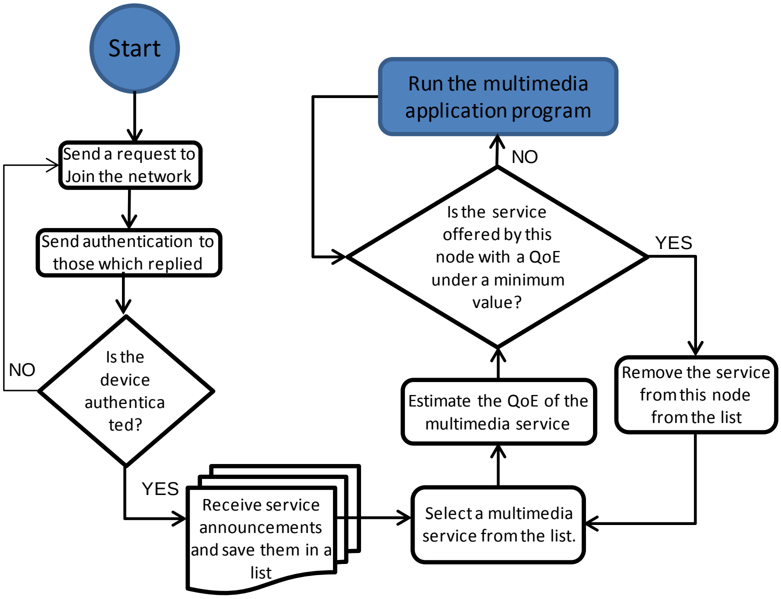

Figure 4.4. Service node selection algorithm.

When a node stops a service or leaves the network, it sends a stop service message to all its neighbours in order to broadcast this change to all nodes and remove the availability of the multimedia service from the network. This information is forwarded through the network using the routing protocol of the ad hoc network.

4.3 MWAHCA: A Multimedia Wireless Ad Hoc Cluster Architecture

Generally, the physical topology of the wireless ad-hoc network is given by the density, the placement of the nodes, and their mobility along the time in the area. They way they are organized in the physical topology and the way they communicate with other nodes determine the logical network topology and the routes followed by the data in the ad hoc network [130]. Once the logical topology is built, end nodes have the needed transport infrastructure for the information exchange. This infrastructure allows monitoring, remote management, data gathering, etc. There could be a huge number of logical topologies depending on the criteria and algorithms followed by the nodes when the network is built [134]. The architecture should take into account the network traffic features with the objective of optimizing the performance and efficiency of the transmissions.

Cluster-based architectures are very common topologies in ad hoc networks. These architectures organize the nodes in small groups of nodes that work independent and autonomously. Nodes in each cluster can communication with other nodes establishing neighborhoods or through the neighborhoods of their neighbors. At the same time, a cluster can communicate with other clusters or with external networks though a higher hierarchical level, which is shared by all clusters. We can find in the related literature many wireless ad hoc clustering algorithms and schemes. The network topology will depend on the neighbor selection criteria when building the cluster. Moreover, the motion of the nodes change the topology constantly, which increases the number of management messages. Neighbor node selection has been widely researched in many network structures [135]. Depending on the purpose nodes can be organized taking into account the geographic distance, decrease the energy consumption [136], decrease convergence time, maximize the whole available bandwidth for data transmission, fault tolerance or load distribution purposes, etc.

Due to the fast grow and development of wireless technologies, wireless ad hoc networks are becoming more and more common between people. There are many emergent applications and new environments to be used. New tendencies are user-oriented and service-oriented wireless ad hoc networks [137], where one of the main ones is the real time audio and video streaming (sometimes their use in other type of networks is very complex or too expensive).

In order to deliver real time data traffic inside a wireless ad hoc network, it is essential to quantify and measure the network Quality of Service (QoS) parameters: Delay, Jitter, Lost Packets and Guaranteed Bandwidth [138]. QoS should be analyzed from two complementary points of views. On one hand, it should be taken into account that for each multimedia stream, QoS parameters should be kept inside a range along the time. On the other hand, different multimedia streams (depending on whether audio or video, or depending on the used codec) will have different optimum QoS values and ranges.

Although there is some interest to manage the traffic flow in ad hoc networks [139], many researchers take into account QoS parameters in their proposals

[140]. But, existing cluster systems do not use QoS parameters as criteria to build the logical topology. So, the obtained QoS values could be outside of the required range. Therefore, the quality of the multimedia communications will not be guaranteed and the Quality of the Experience (QoE) of the end users will be affected. Moreover, QoS parameters should be monitored continuously because their values may vary considerably due to network changes (node joining and leavings, new concurrent audio, video and data streams)

We propose a new architecture to build wireless ad hoc clusters based on QoS parameters criteria. The architecture allows structuring the network taking into account the features of the multimedia streaming, the number of streams delivered through the network and the capacity of the nodes belonging to the network. The objective is to offer a guaranteed and differentiated service for ach multimedia stream in order to optimize the communication between nodes and taking full advantage of the available bandwidth, but guaranteeing the required delay, jitter and packet loss levels.

In this section we detail the proposed architecture to build wireless ad hoc clusters for multimedia streaming with service guaranteed. First we will describe the initial state (Init State) of the architecture, which will be used as a starting point of our protocol. Then, we will define the Multimedia Init Profile (MIP), which collects the multimedia information used by the network nodes. Then, we will detail the system processes for the proper operation of the architecture. Finally, the routing algorithm to estimate the most convenient paths for multimedia communications through the cluster will be explained.

4.3.1 Multimedia Init Profile (MIP)

Let Multimedia Init Profile (MIP) be a data structure which represents the multimedia streams delivered through an ad hoc cluster from a source to a destination node. MIP contains in a single array with all the information needed to decide the route for each stream. It contains the information of the QoS requisites that should be guaranteed by each cluster node to transmit each type of multimedia stream. Network topological features and the capacity of the nodes in the ad hoc network will determine the most adequate number of nodes and the properties of the MIP available to be selected as an initial configuration by a node. The network topological features are the density of the nodes, their location, space distribution (these data is obtained by using GPS data), obstacles and possible signal interferences (estimated by using geographical maps or building maps if it is indoors). Other features such as transmission power and coverage area could also be added in future works. From the multimedia streams we have added the type of multimedia stream (video, audio or both), the used codec and the QoS requisites (delay, jitter, lost packets and Bandwidth).

The cluster-based architecture uses MIP as a main feature to build the clusters. It groups in the same cluster the nodes with the same MIP under the coverage area. We can adapt the definition of the MIP to each particular case. Moreover, we can define several MIPs for a single cluster. For example, a network with low nodes density in the clusters can have both MIP one for audio transmission and other for video transmission, but in a network with high density of nodes dedicated only for the video transmission using many types of codecs, they can use several MIPs that will allow them to transmit the streams with different codecs into different clusters. In order to simplify our explanation and the system deployment details, we will assume that all nodes in the ad hoc multimedia cluster share the same MIP, but it can extended to several MIPs or to MIPs with range of values.

The number of defined MIPS, available to be selected in the system startup, should be wide enough to cover accurately the most common multimedia streams, but it should not be too much to facilitate new node joining and avoid having too many different clusters.

MIP has the following parameters inside: Maximum Bandwidth, Minimum Bandwidth, Maximum Delay, Maximum Jitter, Maximum Packet Loss and Maximum number of Hops. Each MIP has a one byte long hexadecimal code, called HCode, and an alphanumeric code, called ACode, with variable size:

- **Maximum Bandwidth (MaxBW)**. This parameter establishes the maximum bandwidth spent by all the multimedia flows processed by the node at the same time. This value represents the whole bandwitdh provided by the node for multimedia transmissions with guaranteed service.
- **Minimum BandWidth (MinBW).** This value describes the minimum bandwidth required to transmit just one multimedia flow. It represents the bandwidth requirements specified by a multimedia codec or a group of multimedia codecs with similar requirements for a single multimedia communication.

- **Maximum Delay (MaxDelay**). This parameter allows knowing the maximum latency value allowed for a multimedia packet across the cluster between the source node and the target node. It represents the maximum guaranteed quality for audio or video transmissions inside a cluster.
- **Maximum Jitter (MaxJitter).** This value indicates the maximum jitter that is considered tolerable for multimedia transmissions inside the cluster.
- **Maximum Packet Loss (MaxLoss).** This is the maximum percentage of acceptable lost packets. When this value is exceeded, the target node, the node at the end of the multimedia path, breaks the multimedia transmission and notifies the rest of the nodes in the path because the quality of communication cannot be guaranteed. It is calculated to each one individually. This parameter, together with the Maximum Delay and Maximum Jitter represents the quality of service provided for a real-time communication.
- **Maximum Hops (MaxHop).** This value provides the maximum diameter of the cluster and it can be estimated through the routing table. When a new node joins the network, it will start the connection process trying to connect with a cluster using the same MIP. The system uses the MaxHops value to check that the cluster dimension is always kept under acceptable values for multimedia delivery guaranteeing enough quality of service. A node belonging to a cluster will not accept new joining nodes if it has reached MaxHops.

Table 4.1 details the list of MIPs set in advance for audio and video transmission. We have tagged an alphanumeric code to each MIP, called ACode, in order to let the node set each MIP and use it in the protocol messages. It is easy to use by the user or by the network administrator when the initial configuration of the node is set. We have also associated a hexadecimal code called HCode, which has a byte size, which will be used in the protocol header when information is exchanged between nodes in the same cluster. Table 4.1 also shows the QoS parameters associated to each MIP: maximum and minimum bandwidth, delay and jitter, as well as the maximum number of hops (MaxHops) and the cluster diameter.

Table 4.1. Defined MIP list for the practical implementation of the architecture.

MIP	ACode	HCode	MinBW	Max BW	Max Delay	Max Jitter	Max Hops	Max Loss
Audio 32K	A1	0x01	8Kbps	32Kbps	50 ms	20 ms	6	0.5
Audio 64K	A2	0x02	8Kbps	64Kbps	100 ms	40 ms	6	0.5
Audio 128K	A3	0x03	16Kbps	128Kbps	150 ms	40 ms	5	0.5
Audio HQ	A4	0x04	32Kbps	1024Kbps	150 ms	40 ms	4	0.5
Video 256K	V1	0x41	64 Kbps	256 Kbps	100 ms	20 ms	4	1
Video 1024K	V2	0x42	128 Kbps	1024Kbps	150 ms	40 ms	4	1
Video 2048K	V3	0x43	256 Kbps	2048Kbps	200 ms	40 ms	3	1
Video HQ	V4	0x44	1024 Kbps	20Mbps	200 ms	40 ms	2	1
DEFAULT	Default	0xFF	56 Kbps	1 Mbps	200 ms	40 ms	4	1

Figure 4.5 shows an example of a multimedia ad hoc network using a cluster-based architecture. All nodes share an area and all are reachable by the other nodes because they are under their wireless coverage area. They are distributed logically in clusters that are specialized in the transmission of similar multimedia streams with similar audio and video QoS parameters. Figure 4.5 shows how nodes are grouped in 4 clusters, 2 for audio transmission and 2 for video transmission. One video and one audio cluster are dedicated to the transmission of codecs with low bandwidth requirements, the other video and audio cluster are dedicated to the transmission of codecs with higher bandwidth requirements. The head node of each cluster can communicate with head nodes of other clusters in a higher hierarchical level that allows the communication between clusters.

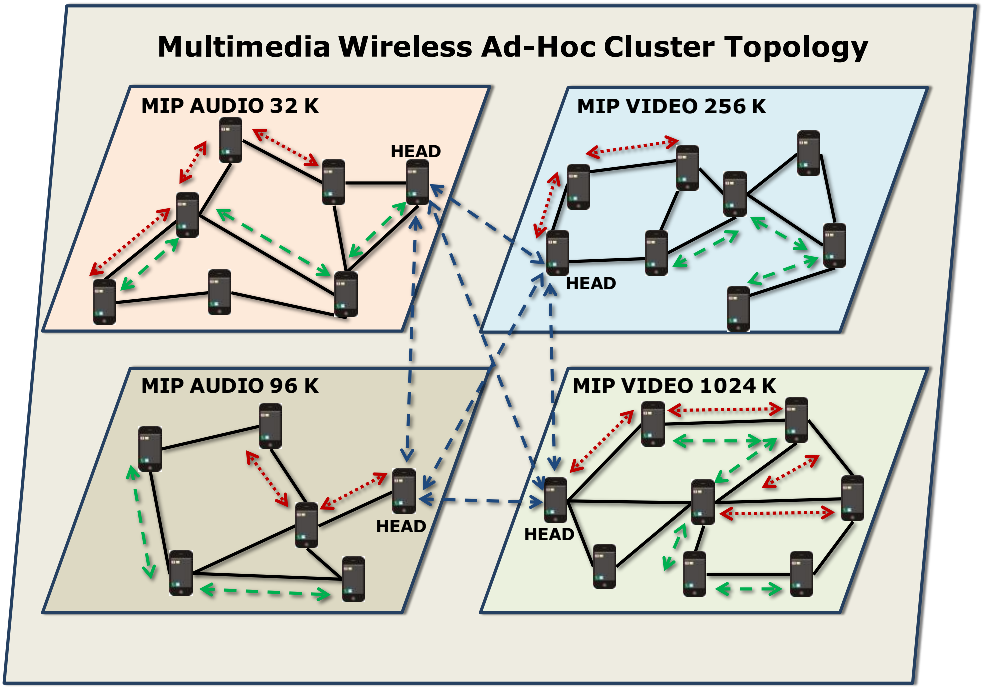

Figure 4.5. Multimedia Ad-hoc Cluster topology

Figure 4.6 shows the elements of the proposed topology and the relationship between them. The architecture has three levels of operation: Hardware Infrastructure, Logic Management and Admin Interface. The hardware infrastructure level is formed by different types of nodes (regular cluster nodes, gateway nodes, and head nodes), which build the physical topology, and clusters, which build the logical topology. When new nodes join the network, they have the regular cluster node role. A regular cluster node cannot communicate with nodes from other clusters or with external devices, but with nodes of the same cluster. When a new regular cluster node tries to join the network, it searches nodes under its coverage area. When it receives replies from nodes having the same MIP, the developed protocol will let them exchange information in order to build clusters following the proposed architecture. Each node in the ad hoc network, despite of its role, can only belong to a single cluster. First node will be the head node, and it will be the responsible to locate and communicate with the head nodes of the other clusters. Gateway nodes have two network interfaces. One interface will be used to connect with the nodes in the ad hoc network and the second interface will be used to connect two with an external network. A node can have both roles: head node and a gateway node.

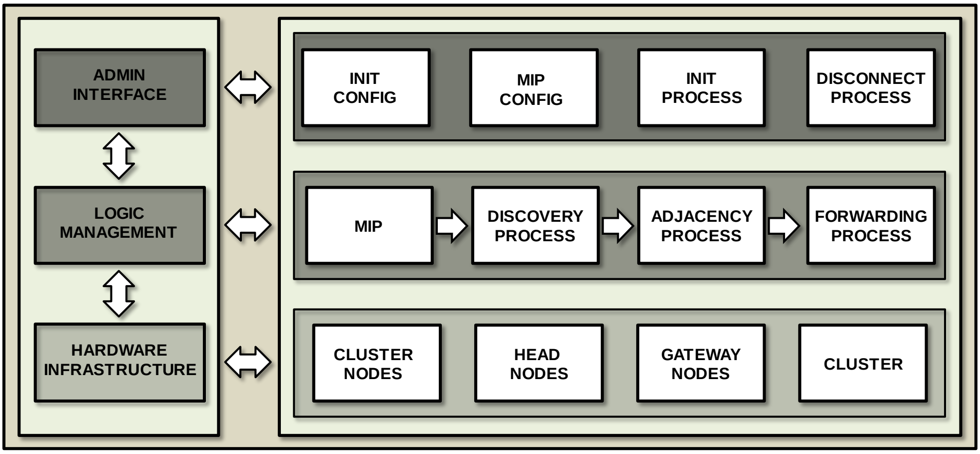

Figure 4.6. Multimedia ad-hoc wireless network architecture elements.

Logic Management level defines the elements of the protocol, which will be used to manage the hardware infrastructure elements by gathering the information obtained from the Admin Interface level. MIP will be used to group the nodes in clusters and assign the cluster to the new nodes. A new node can only be neighbor of a node with the same MIP. All nodes in the same MIP will always have similar features. Figure 4.7 shows the internal organization of a cluster. It is formed by a cluster node and a gateway node that use the same MIP. Multimedia streams can be initiated or ended in external multimedia networks like VoIP, IPTV o ISPs. The connection to external networks is always made by gateway node.

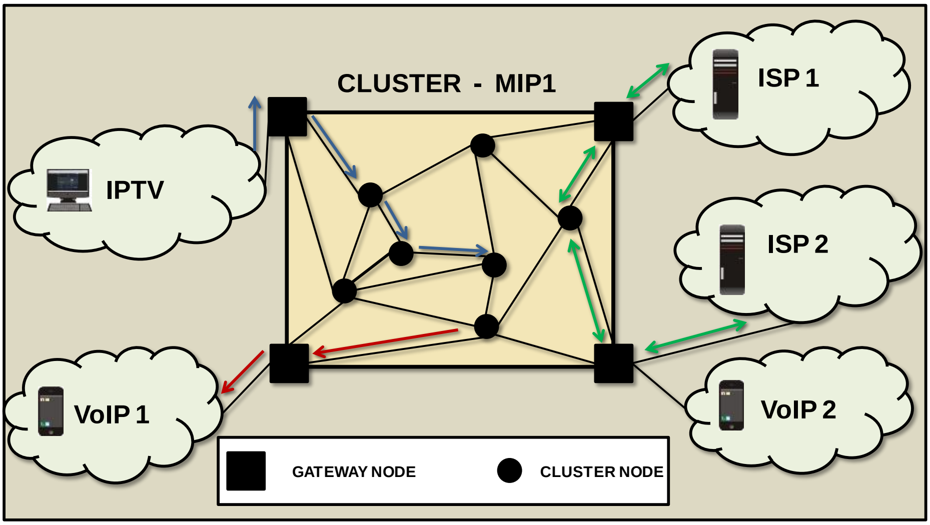

Figure 4.7. MIP Multimedia Cluster

Logic Management level also defines the logical processes performed by the nodes automatically as a function of their states and the events given in the network: Discovery Process, Adjacency Process and Forwarding Process. When a node starts up correctly, it executes the Discovery Process and seeks other nodes with the same MIP under its wireless coverage area. When it finds other nodes, the Adjacency Process starts in order to establish a neighborhood between both nodes. The process is repeated every time it finds a new node with the same MIP, allowing the system to build the network clusters. When a cluster is built, it has the capability and resources to retransmit the multimedia streams which features meet the MIP of the nodes of the cluster. Forwarding process is started when a node starts a new stream query. The query can be started inside the cluster or can be started by other cluster node or by an external network (in this case the query comes from a gateway node). Forwarding Process uses the routing algorithm to know the route that should follow the multimedia stream inside the cluster and requests the resource reservation to each node of the route. This process is responsible for establishing the connection between nodes and to guarantee the QoS required by the MIP during the communication.

The third level is the Admin Interface level, which allows the interaction between the user and the device. By using a Graphical User Interface (GUI), node Init Configuration can be modified, including IP addressing, the MIP to be used by the node, and in case of a Gateway node, the communication between the ad hoc network and the external network. Admin Interface Level is used to manually control the Init Process and the Disconnect Process. The user can initialize the node and join or disconnect the node from the ad hoc network. The node can only be configured before the Init Process starts, so in order to make any change, it is necessary to stop the node, through the disconnect process, perform the appropriate changes, and restart the system with the Init Process.

4.3.2 System Process

In order to design the architecture, we propose four basic processes, which correspond with the basic actions of a node inside the ad hoc network. Each process is associated to a set of states and transitions that will be detailed later when the system state machine is explained. Figure 4.8 shows the relationship between the processes of the system. Init/Disconnect process is the start and end process of the system. It is the only process that requires the user intervention for executing it.

Init Process starts the node when the user (or the system) has selected the appropriate MIP. Disconnect Process allows the user to leave the network safely (or to restart with a new MIP configuration). Init/Disconnect process brings the system to the Discovery process, where the node will try to find the nodes in the network with the same MIP. When the node finds another node with the same MIP, and the cluster does not arrive to the maximum number of hops defined by the MIP, the system starts the Adjacency process, in which both nodes exchange their network information and lets the new node join the cluster. When a node, belonging to a cluster, receives a query for multimedia stream transmission, and checks that it is possible guaranteeing MIP requirements, Forward Process is started.

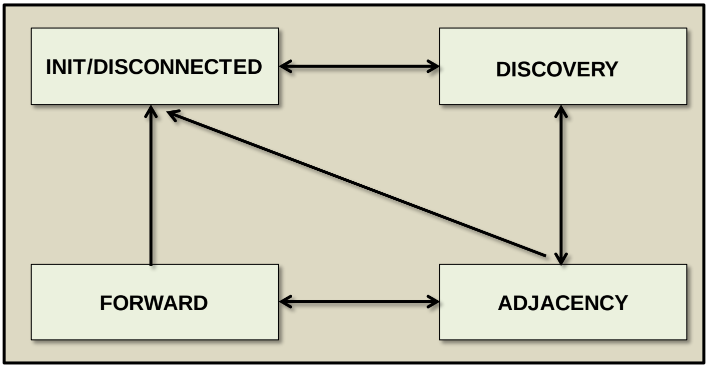

Figure 4.8. System Processes of the multimedia wireless ad-hoc cluster architecture

System Processes, with the states of each process, are described in detail next:

1. **INIT/DISCONNECT PROCESS**: This process includes the subprocess executed by the node when it joins or leaves the network. This system process includes two states: Init state and Disconnect State. The node will be in the init state when it is running the Init Process, and it will be in the disconnect state when it is leaving the network. In the Init Process the node tries to access the physical network and obtain information about possible neighbors. The Init Process is divided into three different phases: MIP selection, unicast IP configuration and group multicast IP configuration. At the first phase, the system allocates the node MIP according to characteristics and the available resources. The MIP of the node can be statically selected by the user, but there is a MIP Default Profile, identified with the HCode value of 0xFF, if no MIP is selected. In the second phase of the Init Process, the IP configuration of the node is established, including unicast IP address, network mask and gateway address. The use of a DNS server is optional and it is not required for the normal operation of the protocol. The IP configuration can be manually configured by the user, or may be dynamically obtained by using IETF Zeroconf as defined in RFC 3927. In the next phase of the initialization process the node joins the multicast group matching its MIP. All nodes sharing the same MIP be listening the same IP multicast group address, thus all nodes in the same multicast group, belong to the same cluster. The range of IP multicast group addresses used by the system is 239.100.100.X/24. In this multicast address, the fourth and last byte

matches the MIP HCode value. For instance, a node with the MIP 256K Video profile with ACode equal to "V1" and a HCode value of 0x41 (decimal value 65) will join the IP multicast group 239.100.100.1.65. Using multicast addresses, each node will communicate exclusively with other nodes with the same profile, without interfering other nodes in other clusters with different MIP. This system process is also in charge of making the node leave the ad hoc cluster it belongs to. This part of the process occurs when the node is in the Disconnect State. The system can reach this state from the other 3 system processes, since the node can leave the system at any time regardless of the assigned state. The node uses the multicast group address of the cluster to notify its neighbors that it is leaving the cluster. Then the neighbors can update their status tables in order to reorganize their Forwarding Process.

2. **DISCOVERY PROCESS:** Upon completion of the init process, the node is ready to make the transition to the Discovery Process. In this process, the node will try to detect the presence of a cluster with the same MIP in order to join it. There are two possible states in the Discovery Process: Discovering State and Stand Alone State. When the node accesses the Discovery Process for the first time, then the system changes to Discovering State. This is an active state, while the node stays in this state, it keeps sending discovery messages to the IP multicast group of its MIP. The node waits 60 seconds for replies after sending each discovery message. If no reply is received, during this time interval, another discovery message is sent. Discovering State has a maximum duration of three minutes. If one or more reply messages are received during these 60 seconds after the discovery message is sent, the system changes to the Adjacency Process. After sending three discovery messages without any result, it changes to the Stand Alone State, but still remains in the Discovery Process in passive mode, that is the Discovery process does not keep sending periodic discovery messages, but the node remains listening for new nodes trying to join the network. If a discovery reply from other node is received, it first checks if the MaxHops of the MIP is not exceeded. If MaxHops is not exceeded, the Adjacency Process starts.

3. **ADJACENCY PROCESS:** This process starts when the above discovery process has detected the presence of one or more nodes with the same MIP. The Adjacency Process includes: Join State, Associated State and Established State. If a node in the Discovery Process receives replies from

two or more nodes belonging to the same cluster it will try to establish the adjacency with all detected nodes. If a node receives replies from two or more nodes belonging to different clusters, but all of them are using the same MIP, the system will choose the best cluster and reject other options. The best cluster choice is made by a three-step algorithm, which uses the information included in the discovery reply messages. First, the node estimates the diameter of the cluster if this neighbor is selected. The best selection is the smallest diameter. In case of a tie, the second step comes. The node checks the number of adjacencies of that neighbor and selects the node with the minimum number of established adjacencies in order to distribute the load between different clusters. Finally, if there is a tie in the previous step, it selects the source node of the first received reply. Once the node selects the best candidate, it sends a Join message to the selected cluster nodes. When a reply message is received, the node changes to the Join state. Then, the new node will receive the information about the cluster characteristics and the topology structure. When the node has the whole information about the cluster, then it changes to the Associated State. Finally, the new node sends the information about its resources and availability to the other nodes in the cluster. When all nodes inside the cluster have the same information, the cluster has converged. Then, the new node changes to the Established State. In this state, the node is fully integrated in the cluster and it is ready for multimedia transmissions. A node remains in the established state indefinitely until it receives a multimedia transmission request, until the user invokes the disconnect process or until the adjacency is broken. When there is a multimedia request, the Forwarding Process starts.

4. **FORWARDING PROCESS:** This process is in charge of the multimedia delivery. Inside the Forwarding process we can find two different states: Queued state and Forwarding state. The Forwarding state can be initiated only when the node has successfully completed at least one valid adjacency with a node. Multimedia requests could be originated by the node, for example a request for audio or video communication performed by the user interface, other adjacent nodes or by a gateway node from external networks. When a multimedia request is processed, regardless of the origin, available bandwidth resources at the node are checked. If the node has enough available resources, the node changes to the Forwarding state, makes a temporary reservation of resources for the transfer and

notifies at the origin node that it is ready for transmission. When all nodes in the path from the source to the target node confirm they have made the resource reservation, the source sends a confirmation message to the nodes in the path to allocate permanently the reserved resources for the multimedia flow. Then, the nodes change their state to the Forwarding state and the multimedia transmission takes place. When a node in the multimedia flow path does not have enough resources for the multimedia connection request (e.g. there is not enough available bandwidth) the node changes to the Queued State. The node in Queued State informs the origin of the multimedia request that it cannot process this request, but it will keep it queued. Then, the source node can wait until the bandwidth resources are released or, if there is some alternative route provided by the routing algorithm, it can cancel the current request and try to establish a new communication using a new path. When the Forwarding Process for multimedia transmission ends successfully, the node changes to the established state.

4.3.3 Routing Algorithm

Source Node (SN) is the node belonging to a cluster that requests a multimedia connection. The request can be performed by a user through the Graphical User Interface or from external networks (in this case the source node is a Gateway Node). Target Node (TN) is the destination node of the multimedia connection, which will receive and process the multimedia streams. It can be a regular node or a Gateway Node.

Every node has its neighbors table, which is built and maintained through the adjacency processes, and the cluster topology database which is built using the topology information received from its neighbors. When a SN starts a multimedia transmission, the routing algorithm uses the multimedia streaming bandwidth requirements and the topology information of the nodes inside the cluster. The estimations to determine the route, including the nodes that will forward the multimedia streams inside the cluster, are performed by the SN. The routing algorithm selects as the first hop the node that it closest (in terms of number of hops) to the TN. When there is a tie, the node with oldest adjacency will be selected. Selected node is called Forward Node (FN). FN will estimate the path to the TN using the same process, so it will obtain the second hop in the route to the TN. This process is repeated till TN is achieved. This information is saved in the

MEDIA_ROUTE parameter, which will be used by the SN in the resource reservation request in order to guarantee the transmission quality. The resource reservation request is firstly sent to the first FN, which will check if it has enough available resources. If it meets the requirements, it uses the information included in the MEDIA_ROUTE parameter of the message to forward it to the next hop. This process is repeated in each node of the route till it reaches the TN. If the TN receives the request, it means that the cluster has enough resources to perform the multimedia communication meeting MIP requisites, so it replies with a confirmation message that will follow the same route in order to confirm the resource reservation in each node. When the confirmation message reaches SN the multimedia communication starts.

In case of not having enough resources when a node belonging to a route does not have enough resources, the request is included in the queue of this node till it has enough available resources. If the SN neither receives a confirmation reply nor a queue request in 30 seconds (for example because a node left the cluster suddenly), it sends a message containing the route verification, which uses MEDIA_ROUTE parameter, to the TN.

Nodes keep updated their neighbor table by sending keepalive messages to all their neighbors and waiting for a reply in less than 10 seconds. If during this process a node detects a topology change, it will send an update message to the rest of the nodes in the cluster to let them update their tables. Both, SN and TN, are able to stop the multimedia streams by closing the communication. They will notify to the rest of the nodes of the route that they have to liberate the reserved resources.

4.3.4 Finite-State Machine

Figure 4.9 shows the system Finite-State machine. We can see its different states and the transitions between states. In this section we describe each state of the system and the conditions and events that will make the node to change from one state to another inside a process.

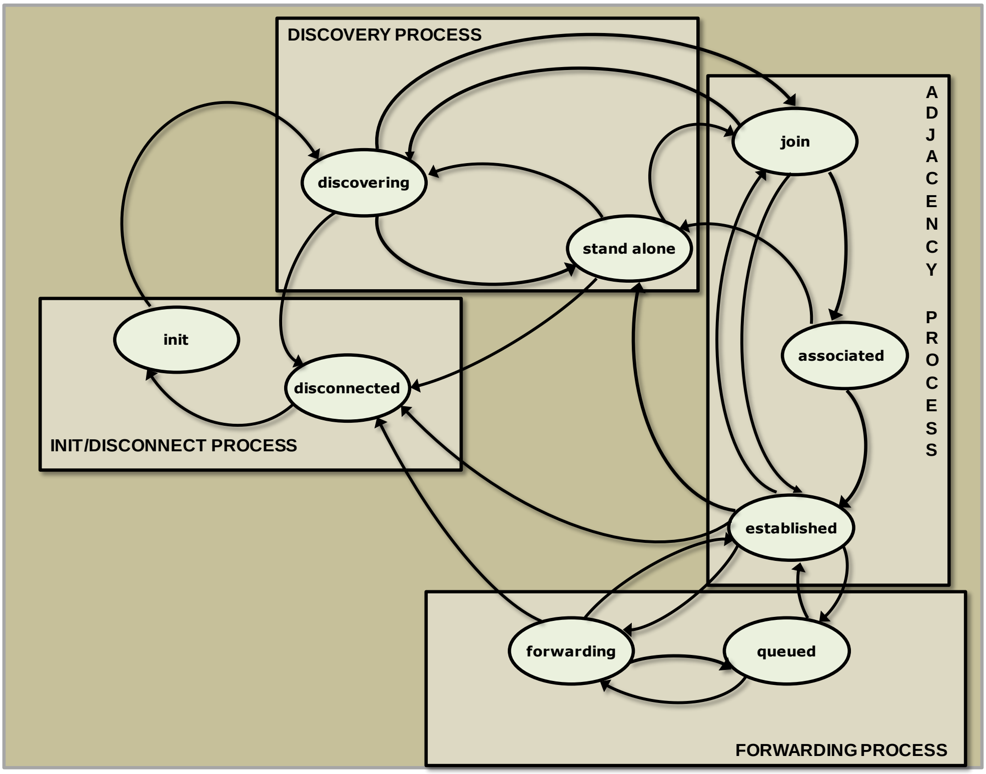

Figure 4.9. Finite-State Machine for Multimedia Wireless ad-hoc cluster architecture

The processes included in figure 4.9 are the following ones:

- **Init State.** This is the initial state of the node during the Init Process. There are two possible ways to access the Init state: First, when the node starts for the first time and, second, when the node is rebooting. There is only one possible transition from the Init state to the Discovering state. This transition is made when the node has initialized correctly, i.e. when the whole information has been obtained from the MIP, the IP settings are correct and the network connection is active. There are several events that may cause the Init process fail: An IP address conflict with another node in the wireless network, the wireless network connection is not enabled, or when it is not possible to join the IP Multicast group. When an error event happens in the boot process, the system remains inactive in the Init state for 120 seconds before it tries again to initialize the system.

- **Discovering State.** In this state the node has not yet established any adjacency and it is looking for a neighbor by sending discovery messages. The first time the system makes the transition to the Discovering state is when, being at the Init state, the system initialization has been completed successfully. The node can also change to the Discovering state from the Stand Alone state. It happens when the system has remained in the Stand Alone state for 12 minutes and no discovery message has been received from other nodes. Finally, there could be a transition to the Discovering state from the Join State when the adjacency fails in the Adjacency process. While the node remains in the Discovering state a discovery message is sent every 60 seconds to the IP multicast address of the MIP. The maximum number of discovery messages is set to 3. From the Discovering state there is a transition to the Join State when the node receives a discovery confirmation message. The waiting time for discovery confirmation messages is set to 60 seconds. Upon finished 60 seconds the node gathers all received messages and process them as explained before. Then, there is a transition to the Join state. After three times of 60 seconds without receiving any discovery message, a transition to the Stand Alone state is made.
- **Stand Alone State.** The node reaches this state when the Discovery process has not found any valid node, and thus cluster, to join. Then, the node remains isolated from the remaining nodes and it does not establish any adjacency. There are two possible ways to arrive to the Stand alone State: First, when the node is in the Discovering state, as described above, and the second is from the Established state (it happens when the node has just one established Adjacency and it is broken because the neighbor node has left the network). A node assumes that its neighbor is down when it receives a leaving notification or when it has not received any response message during 10 seconds (e.g. after a keepalive message has been sent or in the path verification subprocess that takes place in the Forwarding process). There are three admissible transitions from the Stand Alone state: Discovering state, Join state and Disconnecting state. If the node receives a valid discovery message from another node while it stays at the Stand Alone state, then the system replies a discovery confirmation message in order to offer a new adjacency. Then, if it receives a join request message, it will answer with a join acknowledgment message and the system automatically changes the

status to Join state and the Adjacency process starts. If the node remains in the Stand Alone state for 12 minutes and no message has been received from another node, it makes a transition to the Discovering state. Then, the Discovering process starts again an active search for neighbor nodes. Finally, through the intervention of the user, the system can make a transition to the Disconnecting state in order to close the connection and leave the network or to restart because some of the values of the initialization process have been changed.

- **Join State.** This state is the starting point of the Adjacency process. The nodes have not yet shared any information from the neighbor tables but they want to build a new adjacency with the discovered node because it has the same MIP. The system can achieve the Join State from three different states: Discovering state, Stand Alone state and Established state. A transition from the Discovering state is made when the node has received at least a confirmation of the discovery message and the acknowledge join message. The transition from the Stand Alone state takes place when the node has received a new discovery message and a join request message. Finally, the transition from the Established state to Join state occurs when the node has already one or more established adjacencies and it receives a new Discovery message from a new node requesting a new adjacency. The regular next step from the Join state is the Associated state. It occurs when both nodes have exchanged the whole information in its neighbor tables and the routing database. If the transition to the Associated state cannot be completed, because the received information is inconsistent or incomplete, the system will make a transition to the Discovering state (if it is the first adjacency) or to the Established state (if there has other established adjacencies). A node can establish adjacencies with two or more nodes.

- **Associated State.** This is a transient state. Both nodes have exchanged the neighbor tables and the routing database, but they have not yet confirmed the integration of the new node at the cluster. This state is reached from the Join state as described above. From the Associated state the node can make two transitions: Towards the Established state and to the Stand Alone state. The transition to the Established state will occur when the new node receives the cluster acceptance notification. The transition to the Stand Alone state takes place when the node is not accepted and there are no other established adjacencies.

- **Established State.** At this state the node has established at least a valid adjacency and it is integrated inside the cluster. This is the regular operation mode for a cluster node when no multimedia stream is transmitted through the cluster. In the Established state, the node holds a neighbor table with the information about the neighbors and routing database with the cluster topology. The node needs this information to reach other nodes in the cluster and to calculate the best route based on hop count and multimedia available resources. The Established state can be activated by a transition from the Associated state when a new adjacency is established, from the Join state when the an adjacency fails, but there are other active adjacencies in the node, from the Forwarding state when a multimedia communication using that node finishes or from the Queued state, when the resource request remaining in queue is canceled. Possible transitions that can be made from the Established state are to the Join state, when the node receives a new discovery message, to the Stand Alone state, when the last established adjacency in the node is broken, to the Forwarding state, when a multimedia transmission request is received and there are enough resources to process it, to the Queued state when the node receives a request for multimedia transmission but there are not enough resources to process it at that time and, finally, to the Disconnecting state due to the user intervention when he/she wants to disconnect or reboot the node.

- **Forwarding State.** In this state the node is processing and transmitting multimedia packets for every received resource reservation request. This state is reached from the Established state when the first request for resource reservation is received and completed successfully or, from Queued state, when a queued resource request can be satisfied because the node has released enough resources. When the last active multimedia stream on the node finishes its transmission, the system makes a transition to the Established state and it remains listening to new requests. If the node receives a new resource reservation request and the needed resources are not available, then the system changes to the Queued state. Finally, if the user wants to abort the active multimedia connections in the node in order to reboot or to close the node, it makes a transition to the Disconnecting state, but first it notifies it to the Source Node and Target Node of each active communication.

- **Queued State.** The system uses this state when a node is working properly inside the cluster and receives a new multimedia request but it cannot be processed because it has exhausted their bandwidth resources. Queued state can be reached through a transition from the Established state or the Forwarding state when it receives a new stream request. The node leaves the Queued state when it has released enough resources to process the request and it makes a transition to the Forwarding state or Established state. If the resource request is canceled and there are other active multimedia streams on the node, then, a transition to the Forwarding state takes place. But if there are not other multimedia streams processed at the same time, then, the transition is made to the Established state. User can close or restart the node from the Queued state making a transition to the Disconnecting state.
- **Disconnecting State.** The node is in this state when the system is shutting down or rebooting, for example, to update the values of its initial configuration, such as the MIP or the IP settings. The system can change to the Disconnecting state by the user intervention from several states: Discovering state, Stand Alone state, Established state, Queued state and Forwarding State. When the system changes to the Disconnecting state the established adjacencies are checked. If there are adjacencies, a notification message is sent to every neighbor in order let them update their neighbor tables and forward the information to the other nodes in the cluster. If there are active multimedia transmissions, the node notifies the Source Node and the Target Node in order to let them cancel the transmission. If the node is restarting, a transition is made from the Disconnecting State to the Init State.

4.4. Conclusion

This chapter has presented a new multimedia-oriented ad hoc wireless network protocol. It allows sharing multimedia services into the ad hoc network and takes into account QoE and QoS parameters in order to select the best multimedia service provider node. We have shown the designed protocol and the decision algorithms in order to provide the best multimedia service to the end users. On one hand, the system takes into account latency, jitter, lost packets and bandwidth parameters (all of them QoS parameters) in order to select the best service provider node. On the other hand, the system takes into account the estimated QoE parameter (based on a previously studied formula), and the closest node which implies less RTT and thus lower zapping times, in order to have the best QoE.

We have also presented a new cluster-based architecture for ad hoc wireless networks. It uses QoS profiles to optimize multimedia delivery. The architecture provides a flexible solution with the ability to guarantee the quality of multimedia communication over the ad hoc wireless network. It is able to self-adapt to many physical network configurations through the suitable selection of the Multimedia Init Profiles (MIP). The proposed architecture provides a control mechanism to build the appropriate topology for each cluster. Furthermore, the system uses a resource reservation scheme to guarantee the quality of the multimedia streams.

Both sections included in this chapter have been published in international journals with ISI Thomson Impact Factor. They have been listed in the section “publications derived from the PhD”, in the conclusion chapter.

Chapter 5. Fault Tolerance in MWAHCA

5.1 Introduction

It is important to select the appropriate nodes to have connections when there are nodes fail or when a link is lost. The goal of a good implementation is a deployed network with low handover and end-to-end delays for multimedia applications. Packet loss ratios as well as an efficient network selection process parameters must be considered to design a high performance network.

In this chapter, a fault tolerant mechanism for multimedia flows is added to MWAHCA in order to improve the resilience of the designed protocol when a node into the wireless ad hoc network fails. It is based on fast switching paths. The main objective of our proposal is to have an algorithm to provide fault tolerance to each multimedia flow in a wireless ad hoc network running a protocol based in the MWAHCA architecture. We are not going to modify the original architecture but we are going to propose a protocol enhancement and extension. In order to avoid the degradation of QoS parameters such like latency, jitter or packet lost, the algorithm looks for a fast switching path while the system is being restored or a new optimal path is been recalculated and guaranteed. The fast switching path introduced by the algorithm is not a guaranteed route and it will be dismounted when an optimal path is found. The main objective of this proposal is to reduce the convergence time when a network failure happens and the end user could keep receiving multimedia packets as soon as possible after the network failure.

5.2 Adding fault tolerance to the MWAHCA architecture

In the MWAHCA architecture, nodes are grouped in several clusters following the MIP configuration as described in a previous section. The MIP configuration collects the hardware characteristics that a cluster node must fulfill to join other cluster nodes in order to complete an adjacency. Following the reference model, only adjacencies between nodes with the same MIP are allowed, thus each cluster in the network is created by joining nodes with similar behavior and available resources. MIP configuration is initially assigned to cluster nodes. It limits the types of multimedia flows that can be managed by a specific cluster node. In this subsection, we are going to focus only on the fault tolerance mechanism based in fast flow recovery, but we are not going to deep inside the whole network structure or in the different process to make or destroy an adjacency. The starting point of this research is a multimedia wireless ad hoc network where at least one cluster has already been created. This wireless ad hoc cluster provides us the transport infrastructure to deliver multimedia data from a source node to a destination node. They can be both internal and external to the ad hoc wireless network. Figure 5.1 describes the components of this scenario. Gateway Nodes (GN) are defined in the MWAHCA architecture to be used as a bridge between two different networks or with external nodes. A GN is connected with the wireless ad hoc network by a wireless interface card, but they are also connected to another external network with a different interface card. This external network can be wired or wireless, but in this research we only use the wireless connections option.

When a cluster node has to transmit multimedia packets to another cluster node, it looks for the best route according to the routing algorithm provided by the MWAHCA architecture. This algorithm examines the whole cluster topology, which is defined by the assigned MIP, and it weights up the resources and characteristics of the nodes into the path in order to select the optimal multimedia route. When the best path is calculated, then, the required resources for the multimedia streaming are reserved by exchanging messages between the source and destination nodes, before starting sending multimedia packets. All nodes in the path are checked to guarantee that they have enough capacity and no more multimedia connections are allowed in a node when bandwidth resources run out.

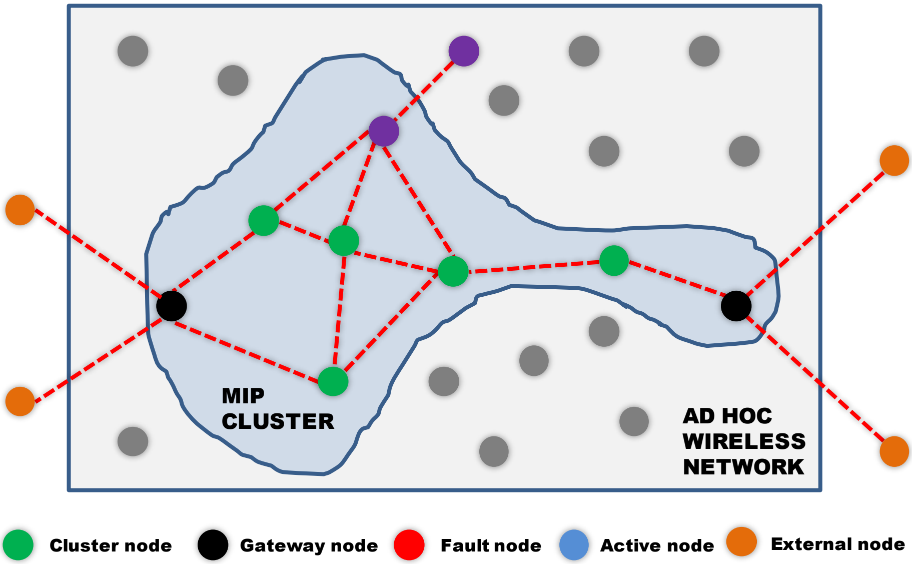

Figure 5.1. MIP cluster built into the wireless ad hoc network

If the rouging algorithm offers different paths between source and destination nodes, only the best route will be selected according to the topology characteristics and the multimedia metrics. When a node on the selected route cannot make an appropriate resource reservation to accomplish the request from the source node, it sends a notification message back to the source node, thus it could eliminate this saturatted node from the topology used by the algorithm to avoid bottlenecks and traffic congestion. Then, a new path is calculated and resource allocation is tested again. Because routing algorithm works independently for each flow, this mechanism provides load balancing between two nodes into the same cluster. Sending different multimedia flows between two nodes by different routes keeps latency, jitter, and lost packets under the thresholds criteria established by the MIP configuration.

When the resource allocation is successfully completed and multimedia streams are sent through the same path and the same nodes, thus jitter and latency will be the lowest because of the routing algorithm calculations. But, the problem here is when a node in the path is disconnected, failed down or when it just disappears from the cluster. The neighbor nodes will detect the fault node when it stops sending messages. Then, they update their state tables and send this change of topology to other nodes in the same cluster. When the source node finds out that there is a multimedia path that is not valid anymore because of a change in the topology, it stops sending multimedia packets and runs the routing algorithm again. If there is an alternative path, and the nodes meet the requirements, then a new route is built between the source and destination nodes. Consequently, the MWAHCA architecture can be considered inherently fault tolerant because of its capacity to adapt when changes happen. However, the architecture does not provide any specific fault tolerant mechanism, thus the convergence time needed to recover the multimedia flow can be high enough to break the multimedia communication. Even, when the transmission is kept, the QoS parameters and the quality perceived by the end user will be really harmed in this time interval.

The convergence time depends on many factors: topology size, amount of nodes into the cluster, average adjacencies by a node, average distance between nodes, and node settings such as timers. From previous studies performed by us through simulations and real scenarios, the convergence time can be estimated and it takes values in a wide range of values, on general between one and sixty seconds. For some applications, such as IPTV o VoD, convergence time values around a few seconds can be admissible because the multimedia transmission will be lost for a while but it will be restored quickly. A fault node will cause some sort of degradation in the quality of service perceived by the final user, but the transmission will not be interrupted. Nevertheless, if there is a real time bidirectional communication, like a VoIP call or a video conference between two or more people, and the destination node stops receiving the audio or video signal only a few seconds, the communication breaks down. Moreover, the user may end the transmission because he/she cannot hear or see anything.

Another different problem that we can face with this fault tolerance behavior is the suboptimal routing. If a new route is calculated and the multimedia transmission is restored, when the failed node restarts and it comes back to the cluster, the best path will not be calculated again and the available resources can be underused.

The related problems with the original behavior bring us to design a new complementary algorithm valid for any wireless ad hoc network in general and for MIP clusters based on the MWAHCA model in particular. In the next subsection this proposal is detailed.

5.3 Fault tolerant mechanism based on a fast switching path

Our objective is to design a fault tolerant mechanism that should be able to restore a multimedia transmission path in less than a second. Moreover, this mechanism will be applied to individual unidirectional flows, thus if two or more flows are affected by a node failure, each one of them will be recovered separately. A bidirectional multimedia communication, VoIP call o video conference, is seen by this algorithm as two different unidirectional and independent flows. To achieve this objective, the algorithm establishes a temporary route called *Fast Switching Path (FSP)*. A specific FSP will be assigned to each affected flow to quickly switch from the broken path to a new optimal route when it is calculated. The purpose of this FSP is to reduce the convergence time, but at the same time to keep QoS parameters inside an admissible range of values. Multimedia transmissions usually use UDP protocol at the transport layer, thus when a node fails, it is not possible to completely avoid the packet loss because the acknowledge packets are not sent to confirm the correct packet reception. Hence, even with convergence time values lower than a second, some quality degradation will be always expected.

When a node is transmitting multimedia packets and the next hop in the path stops sending messages back to it, the node becomes an *active node (AN)*. On one hand, the role of an active node seems similar to a source node, because it has to run the routing algorithm to calculate the best path from this point to the destination. But, on the other hand, an active node does not make a reservation because there is already a multimedia transmission in progress and it needs to restart sending packets as soon as possible. When the algorithm determines the new next hop to the destination node, then the active node sends a *Warning Message* addressed to the destination through this hop. Each hop between the active and the destination node that reads the *warning message,* updates its information table and forwards the message to the new hop. The *warning message* is only used to advise the nodes about the new path to retransmit the packets as long as the resource starvation does not happen. Consequently, in the first stage of the fault tolerant mechanism, there will be two concatenated paths for the same multimedia flow in the same cluster. First path is part of the original broken route and it goes from the source to the active node; this is a guaranteed path because all nodes keep the resource allocation initially performed. The second path goes from the active to the destination node and it is the fast switching path. FSP does not have any guarantee of service and packet loss may happen, but destination will keep receiving packets just some milliseconds after the node failure has been occurred.

To easy understand how the proposed mechanism works, we are going to study an example topology. Figure 5.2 shows two multimedia transmissions along the same MIP cluster. In this example topology there are twelve nodes, two of them are gateway nodes and they allow the ad hoc wireless network to connect with external nodes. The system is designed to provide multimedia transmissions between internal sensor nodes, between internal and external nodes or vice versa, or between external nodes. In this example we will describe the worst case, between external nodes. So, Figure 5.2 shows how multimedia transmissions go from the left to the right external nodes. The wireless ad hoc network is only used as a transport network between external nodes. Thus, all the multimedia traffic that will be considered in this example crosses the ad hoc network from the gateway node 1 (GW1) to the gateway node 2 (GW2) because only a unidirectional flow has been taken into account. Some multimedia applications require bidirectional flows; however, as the fault tolerant mechanism is applied independently to unidirectional flows, this scenario holds all the necessary components to understand how it works. Moreover, it could represent a real case study where the wireless ad hoc network is used to deploy a multimedia service, extending their scope and coverage. Figure 5.2 also shows the best path calculated by the routing protocol for both multimedia flows when all nodes into the cluster are working properly, the adjacency processes have finished and the network has converged. All nodes in the ad hoc network have the same characteristics, the same available resources and they have been previously configured with the same MIP settings. Therefore, the main criterion used by the routing algorithm is to minimize the number of hops between GW1 and GW2.

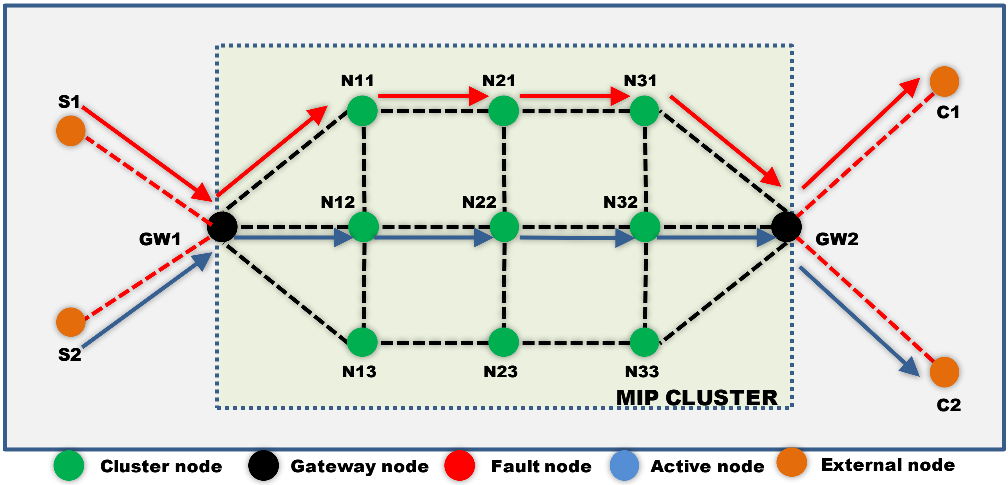

Figure 5.2. Two multimedia flows crossing along an ad hoc network using the optimal path

Figure 5.3 shows the FSP calculated for the red flow when N31 node fails down. In the described scenario a node failure is simulated by shutting down the N31 node (the red node in the diagram). In a certain moment, packets of the red flow transmitted by the N21 node are lost. When N21 node breaks the adjacency with N31 node, it sends a *Cluster State Update (CSU) message* to the remaining neighbor nodes. Next, with the purpose of restoring the multimedia transmission quickly, the active node starts the fault tolerant mechanism running the routing algorithm to find out the FSP. Looking at the diagram, we can see that the best option to get to the destination is using N22 as next hop. Then, N21 sends to this node a *warning message* and the N22 node will retransmit the *warning message* to the N32 node, and this one to the GW2 node, who is the final destination inside the MIP cluster. Immediately after the active node has sent the *warning message*, multimedia packets transmission for the red flow is started again, but now it is done through the N22 node. The established fast switching path allows the system to keep delivering packets to the destination, while minimizing the number of lost packets and the amount of time that the service was interrupted because of the node failure. Fast switching moves the system to a transitional state, where the main objective is to provide a continuous delivery of packets although the quality of service of these packets cannot be completely guaranteed. In this transitional path, as observed in figure 5.3, the path between source and destination gateway nodes is not the optimal route because it has been modified by the active node to avoid the failed node. Furthermore, another inconvenient for this transitional route is that the restored multimedia flow is crossing several cluster nodes that are already managing some other multimedia flows, like the blue flow. Thus, if the fast switching path is kept during a long time, it can contribute to make a bottleneck and the network could become congested. In the topology shown at Figure 5.3, red and blue flows are being retransmitted through N22, N32 and GW2 nodes. This is the optimal route only for the blue flow which also has a valid resource reservation in these nodes. But red flow does not have any reservation between the N21 and the GW2 node, thus when one of these nodes runs out their bandwidth resources, the red multimedia packets will be dropped. Blue multimedia packets will not be harmed because of the resource reservation for this multimedia flow is kept at all times.

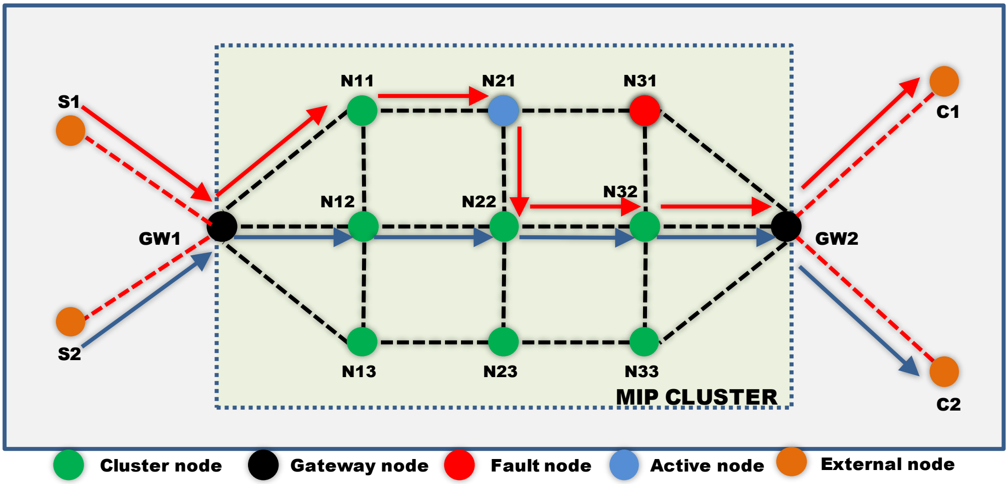

Figure 5.3. Fast Switching Path (FSP) recovery after a node failure.

Given the aforementioned reasons the transitional path should be changed as soon as possible to a new stable path. In order to know the optimal route, the path must be calculated at the source node of this flow (or the first hop in the MIP cluster if the source node is an external node), and not in the active node. It can be be done, because the active node sends a *fault message* back to the source gateway node asking for a new route calculation, but in this process multimedia packets keep being sent through the transitional route provided by the FSP. Moreover, when the source gateway node receives the *fault message* it keeps sending multimedia packets through the current route until a new route is calculated by the routing algorithm and the resource reservation of this final route is performed. Figure 5.4 shows the new optimal route, called *restored path,* for the red multimedia flow, where the number of hops has been minimized (underused nodes were preferred). For the nodes in this new route, a normal resource allocation has been performed following the MWAHCA architecture specifications and the original developed protocol. Finally, when the resource reservation for the restored path is completely confirmed, the source gateway node stops sending red multimedia packets over the FSP and starts sending them along the restored path. Additionally, a message is sent to the active node to release the assigned resources in the nodes in this path.

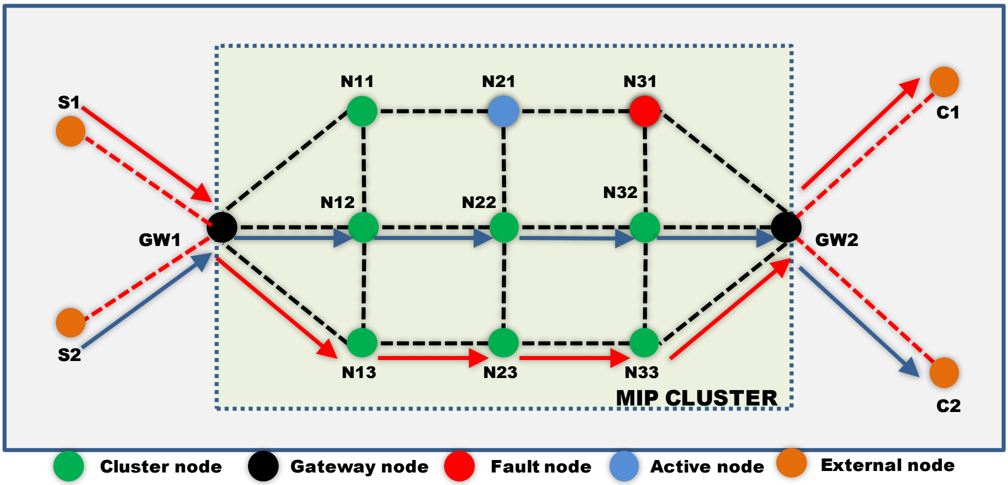

Figure 5.4. Restored path after the node failure

5.4 Protocol Extensions

With the proposed fault tolerance mechanism, MWAHCA architecture has not been modified and the processes and messages defined at the network protocol remain identical. This was one of the initial objectives of this proposal. The other one was to reduce the recovering time when a node fails. It will be demonstrated later on a theoretical study first and in a performance study in a laboratory test bench.

The fault tolerant mechanism can be incorporated to the existing system by an extension of the protocol. To achieve it, we need to define several new types of messages and some messages exchange processes. Table 5.1 shows the new types of messages definition for the fault tolerant mechanism.

Table 5.1. Fault Tolerant Messages

FAULT TOLERANCE MECHANISM				
MESSAGE	**TYPE**	**LENGTH**	**VALUE**	**DESCRIPTION**
WARNING	0x41	13 – 41 Bytes	NODE_ID MEDIA_INFO	Message to inform the next nodes that a multimedia flow without guaranteed services is going to be sent
ACK WARNING	0x42	13 – 41 Bytes	NODE_ID MEDIA_INFO	Confirms the warning message when there is a valid route from this point to the destination
REJECT WARNING	0x43	13 – 41 Bytes	NODE_ID MEDIA_INFO	Reject the warning message when there is not any valid route from this point to the destination
END_FAULT_TRAN	0x44	13 – 41 Bytes	NODE_ID MEDIA_INFO	Releases resource reservation and notifies the previous hop in the path when a node fails
NODE FAULT	0x45	13 – 41 Bytes	NODE_ID MEDIA_INFO	Notification of a node failure from the active node to the source node
ACK NODE FAULT	0x46	13 – 41 Bytes	NODE_ID MEDIA_INFO	Confirm the node failure message sent to the active node

MEDIA_INFO structure holds information about the source and destination multimedia flow, the required resources and the original path. Figure 5.5 shows the messages exchange taken place between the affected nodes when a specific node fails in the cluster. When the routing algorithm selects the neighbor node, only the *warning message* is sent before the multimedia streaming packets are sent again from this new path. If the neighbor node has a valid route to the final destination, and this route does not include the active node, then it sends an *ACK warning message* to the active node to confirm the new route. When this happens, the active node will keep sending multimedia packets. The active node does not wait to the *ACK warning message* reception, thus the latency time for the buffered packets in active node is significantly reduced. The same process is repeated in every hop until the destination GW is reached. The neighbor, intermediate and destination gateway nodes keep the flow information held into the MEDIA INFO structure in a separated table of flows with a valid resource reservation, therefore, when the node becomes congested, it starts dropping packets from no guaranteed flows. When the gateway destination node starts receiving multimedia packets again, it stops the oldest path because it is not valid anymore. The *end fault transmission message* is used instead of the original *end transmission message* in order to notify the nodes in the path that the transmission is ending because of a node failure.

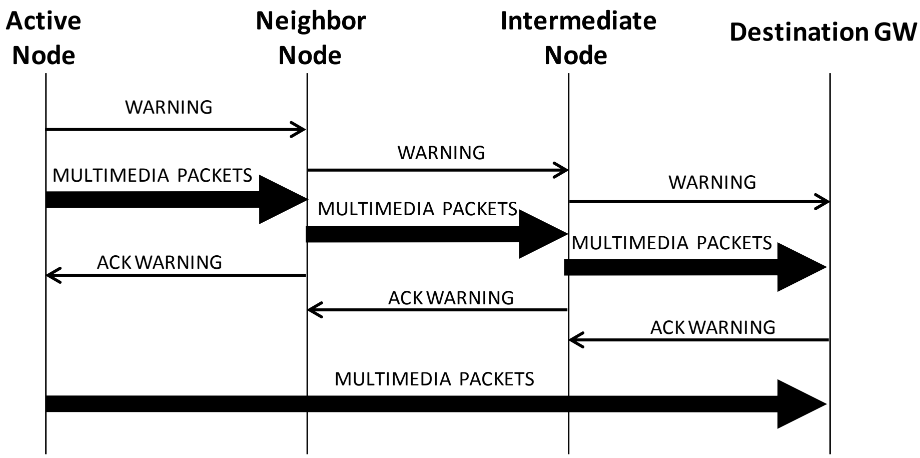

Figure 5.5. Fast switching path stage messages exchange

Figure 5.6 shows the messages exchange when a node in the FSP does not have enough available resources because it runs out its bandwidth. This mechanism is implemented to avoid the bottle neck formation. The intermediate node, which has its resources exhausted, stops forwarding the warning message to the next hops. Instead of that, it answers with a reject warning message. Then, the previous node stops forwarding multimedia packets and forwards the reject warning message back to the path. When the reject message arrives to the active node, the multimedia packet stops until a new alternative FSP is calculated by the routing algorithm. If there is no valid alternative FSP from the active node to the destination node, the active node definitely ends the multimedia transmission by sending an end fault transmission message to the source node by using the original optimal path. In this case, the source node starts the forward process to find a new route to the destination

As stated above, the new FSP is created only to keep the quality of service parameters as good as possible, but the active node has to inform the source gateway node that the current path has failed and a new optimal route has to be recalculated. A *node fault message* is sent from the active node to the source node with the MEDIA_INFO structure of the multimedia flow that must be restored. Figure 5.7 describes this scenario where the source GW node finds a new optimal alternative route to the destination GW and immediately starts transmitting multimedia packets. This new path will be a guaranteed route because all intermediate nodes will have confirmed the resource reservation with a *reserve resources message.* The source GW node will send an *ACK node fault message* to the active node in order to notify it that the temporary fast switching route should be disassembled

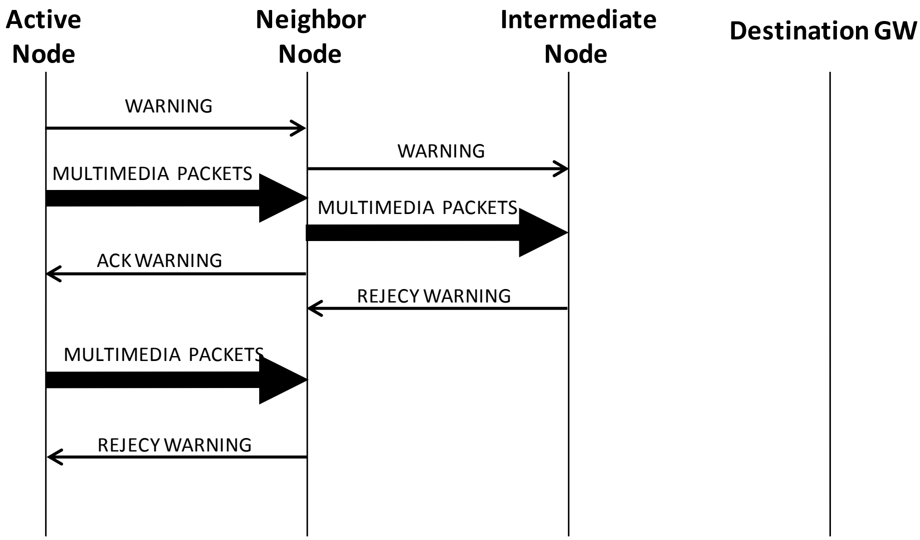

Figure 5.6. Fast switching path (FSP) fails because of resource starvation

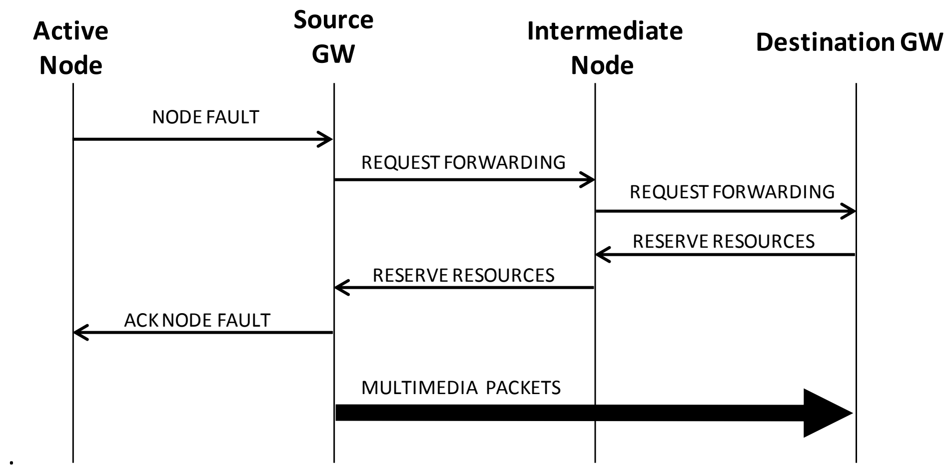

Figure 5.7. Recovery route procedure

When the active node receives the *ACK node fault message,* it stops forwarding multimedia packets and it sends an *end fault tran* to the destination node in order to release the flow information from their tables. Figure 5.8 shows how the fast switching path is unmounted.

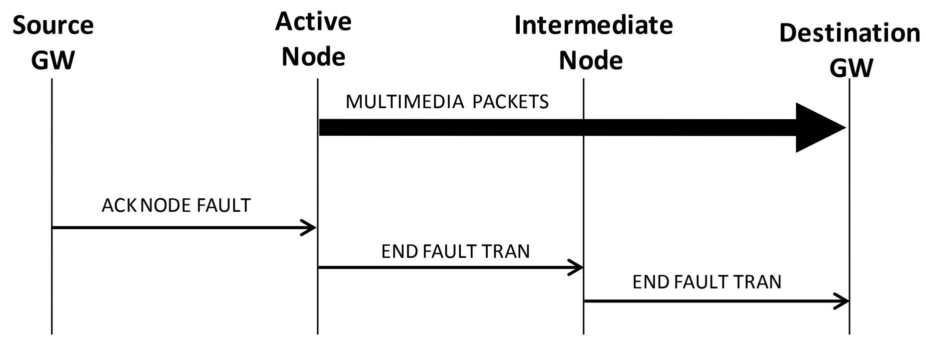

Figure 5.8. Procedure of resource release on a failed path

5.5 Analytical Model of Convergence Time and QoS Parameters

In this section, the convergence time for the fault tolerant mechanism described above is described analytically. Then, the quality of service parameters are analyzed to know the impact of a failed node in the quality of multimedia transmission perceived by the end user.

We define the convergence time (T) as the time interval between a node into the cluster fails and a new route to the destination is established. We assume n_a hops between the source node into the MIP cluster and the active node, the node before the failed node. The latency time to the active node (L_a) is defined as the time needed by a message sent by the source node to reach the active node. The latency between the source node and the destination node into the cluster (with a distance of nd hops) is defined as L_d. The time spent sending a warning message from the active node. is defined as L_w. Then, we can estimate the convergence time from two conditions: when the fault tolerant mechanism algorithm is not used, T1 (given by equation 5.1), and when it is used, T2 (given by equation 5.2).

$$T_1 \geq L_a + 2L_d \tag{5.1}$$

$$T_2 \geq L_w \tag{5.2}$$

If we assume that the processing time is the same in both the source and active node and we make the approximation of taking the latency as a linear function of the number of hops, we can express $L_a = n_a \cdot L$, $L_d = n_d \cdot L$ and $L_w = L$, where L is the delay introduced by one single hop. Acording to these approximations, we can calculate the difference between the convergence time in the two considered conditions. It is estimated from (5.1) and (5.2):

$$\Delta T = T_1 - T_2 \approx L_a + 2L_d - L_w = (n_a + 2n_d - 1)L \tag{5.3}$$

If we call R the average number of packets sent by a specific multimedia flow per second, then we can calculate the average packet loss (P) in the two aforementioned conditions:

$$P_1 \geq (L_a + 2L_d)R \tag{5.4}$$

$$P_2 \geq RL_w \tag{5.5}$$

And following the same approximations than in (5.3), the difference between lost packets when we use the fault tolerant mechanism is given by:

$$\Delta P = P - P_2 \approx (L_a + 2L_d)R - RL_w = (n_a + 2n_d - 1)RL \tag{5.6}$$

If we take the example described in figures 5.3 and 5.4, then na=2 and nd=4 and we assume R=400, then, the convergence time and number of lost packets can be represent in function of L. Figures 5.9 and 5.10 show the linear funcion of ΔT and ΔP when L rises. The improvement achieved on reducing the convergence time and the number of packets will be bigger when more delay introduces a single node. Thus, results can be especially interesting on WSN where nodes have really limited resources.

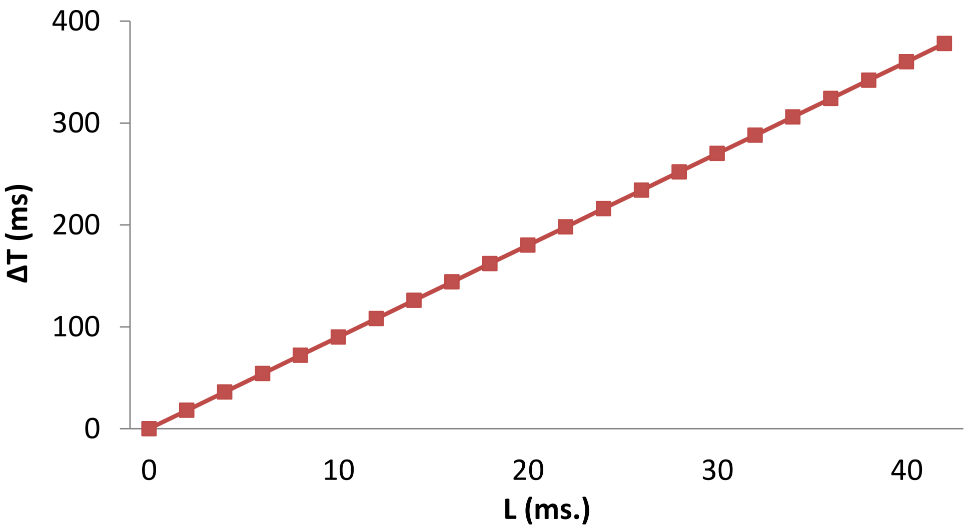

Figure 5.9. ΔT as function of L

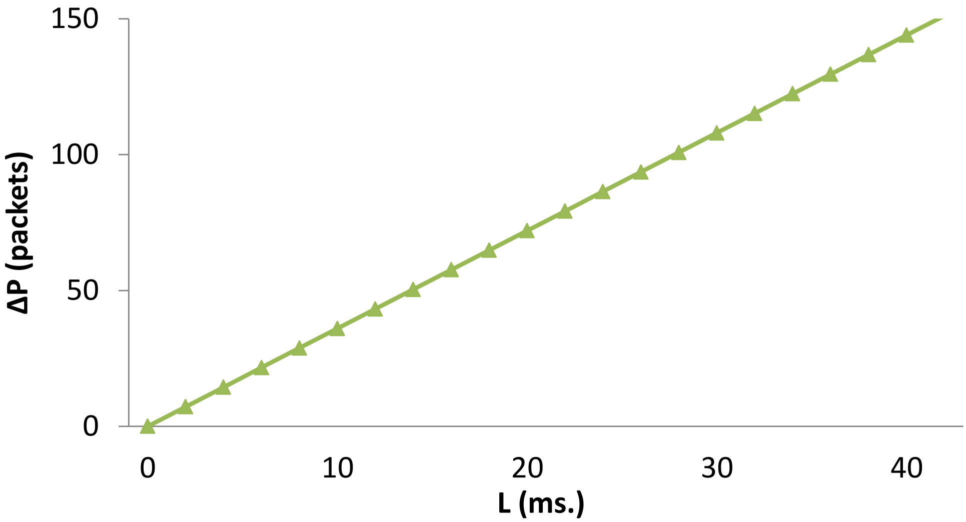

Figure 5.10. ΔP as function of L

5.6 Conclusion

In this section, we have designed and included fault tolerance characteristics to MWAHCA architecture and protocol. Several shortcomings have been detected, thus a new fault tolerant algorithm has been proposed to be applied over ad hoc wireless networks. This mechanism uses a temporary *fast switching path* to improve the QoS and QoE parameter in the recovery transition. In this process multimedia packets are forwarded through a new route, which may not be the optimal route and some nodes on the path have not properly performed a resource reservation. With the new fault tolerant mechanism, convergence time is significantly reduced and the amount of lost packets is minimized. While multimedia packets keep being forwarded through the fast switching path, the fault tolerancet mechanism builds a third and optimal path, the *recovered path,* in order to improve other QoS parameters such as latency or jitter. The last path is built following the original forwarding process provided by the MWAHCA architecture; therefore resource reservation is made in all nodes on the path. Moreover, results in the laboratory show that the fault tolerant algorithm works as expected. It allows a multimedia flow to keep crossing the ad hoc network with a minimum convergence time after a node fails.

The content of this chapter has been published in an international journal. It is listed in the section “publications derived from the PhD”, in the conclusion chapter.

Chapter 6. Architecture Optimization for Wireless Sensor Networks

6.1 Introduction

The number of Wireless Sensor Network (WSN) real deployments is increasing considerably in the last years. It is mainly given by their huge benefits [141]. New wireless technology standards, recent advances in energy-efficient hardware and video coding algorithms are allowing multimedia delivery over ad hoc networks. Nowadays, the features of the sensor nodes and smart devices are very similar to the regular personal computer features. Last generation of sensor nodes can include advanced models of CPUs, with several cores, 1 or 2 GB of RAM, and storage capacities up to 64 GB. Moreover, they can include multiple wireless interfaces such as Bluetooth, Wi-Fi, 3G, 4G, etc.

The amount of multimedia services that can be offered through the network are very large [137][142]: VoIP, IPTV, radio, teaching, multimedia streaming, games, etc. There are a lot of multimedia platforms and protocols used in different fields [126], from the entertainment to the training in the business environment. Multimedia ad hoc networks can be ideal to allow a distributed multimedia service in commercial and social environments that require high visibility to the offered products.

With the widespread use of wireless technology, the ability of mobile wireless ad hoc networks to support multimedia services with Quality of Service (QoS) has become a challenging research subject as described by Khoukhi et al. in [143]. Due to the severe limitations of the ad hoc networks (in terms of energy, processing power, memory, bandwidth, etc.), it is necessary to carefully design the multimedia ad hoc network protocol. Some works are focused on proposing multichannel cross-layer architectures [144], while others are focused on providing fast rerouting algorithms [145], but in this case we focus our research on providing the best topological structure based on the type of multimedia streams.

There have been many studies proposing different topological structures for ad hoc networks that can be summarized into two main types: planar and hierarchical topologies [146]. Planar topologies in ad hoc networks may be of great complexity, mainly in mobile ad hoc networks, because any node displacement may change the entire network topology. For this reason most of researchers have proposed the use of a hierarchical structure for performing an ad hoc network topology [134]. In many cases this hierarchical structure split nodes into different groups called clusters [136].

In this chapter, we show the modification of MWAHCA protocol to meet the wireless sensor networks characteristics. It allows us to have a new multimedia protocol which takes into account the QoS in WSNs. The protocol uses the QoS parameters to structure the network topology. Then, a node decides where to join based on its QoS needs. Header fields, message tables, processes and the data structure managed are the elements of the protocol developed in this section. Basically, we have particularized MWAHCA to wireless sensor networks and we have focused this work to the design and deployment of the network protocol.

First, we describe the architecture features, the elements of the framework and their relationship. Then, we explain the characteristics of the protocol, the structure of the protocol header and the protocol fields. Finally, we show the messages designed for the proper operation of our proposal. The main objective of the protocol is to let the sensor nodes communicate taking into account multimedia flow characteristics. It uses a cluster-based ad hoc architecture that will control the QoS parameters for each multimedia communication, by establishing the appropriated values and guaranteeing the service along the time. The protocol allows the sensor communicate and exchange information about their state and properties. Moreover, sensor nodes use this information to determine which the most appropriate neighbors to be selected are. The protocol dynamically manages the creation of the cluster as a function of the network features, the number of devices, the sensor capacity and the multimedia flows.

6.2 System Architecture

The starting point of our system is a set of wireless sensor nodes located in a delimited place which form a WSN. Each wireless sensor node has different power, processing, memory and transmission capacities. They are able to select other wireless sensor nodes as ad hoc neighbors if they are under their radio coverage area. Wireless sensor nodes are responsible of retransmitting the multimedia flows, which may be audio or video, and can use a wide range of codecs.

Figure 6.1 shows the elements of a cluster. Some sensor nodes are able to provide sensed data to the WSN as Audio IP or Video IP services. There are three types of communications as a function of the source or destination of the communication:

1) Communication started from outside the WSN to a node placed inside the WSN

2) Communication started from a node placed inside the WSN to a destination outside the WSN

3) Communication started from a node placed inside the WSN to a node of the WSN. A node from an external network can provide multimedia contents and audio and video real-time communication services.

Wireless sensor nodes can sense multimedia data or act as a data forwarding nodes inside the WSN. They can communicate with other nodes under their coverage area. New nodes will select the better reachable cluster based on their features and the type of multimedia traffic that is going to be transmitted. We can distinguish in Figure 6.1 two types of nodes, sensor nodes that do not have any connection with nodes from an external network (they can only establish connections with nodes from their cluster), and sensor nodes that have connections with other clusters (cluster heads) or with an external network (gateway nodes). Gateway nodes have two interfaces at least in order to connect with the WSN and with the external network.

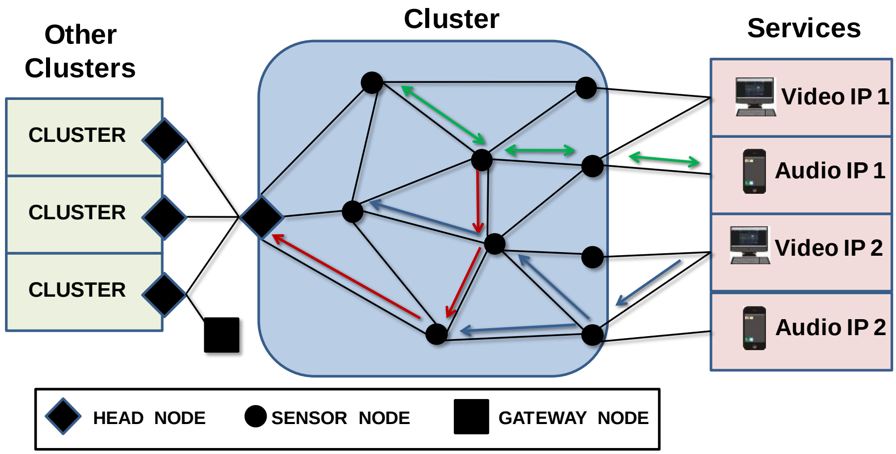

Figure 6.1. Cluster elements and possible communications.

The network is organized in clusters. Every cluster of the architecture is dedicated to a specific multimedia flow, which will be identified by predefined multimedia profiles. We have created a Multimedia Init Profile (MIP) in order to manage the configuration of the sensor nodes [59]. MIP defines which type of multimedia flow can be delivered by the sensor node. MIP groups in a single logical component all required information to guarantee the adequate QoS to the multimedia traffic. MIP gathers the restrictions that will be applied to QoS parameters for the multimedia flows (Bandwidth, Delay, Jitter and Lost Packets) and the cluster properties (maximum number of hops and the number of connections with external networks). There is only one MIP associated to each cluster in the WSN, but there could be several clusters using the same MIP. When a MIP is assigned to a sensor node, the following information is assigned: type of multimedia traffic (audio or video), range of codecs that can be used by the multimedia flows inside the cluster, maximum bandwidth available for retransmissions, and the maximum admissible Delay, Jitter and Lost Packets.

Figure 6.2 shows the elements of the proposed architecture, and their relationship. The architecture defines three operation levels: *Hardware Infrastructure*, *Logic Management* and *Admin Interface*.

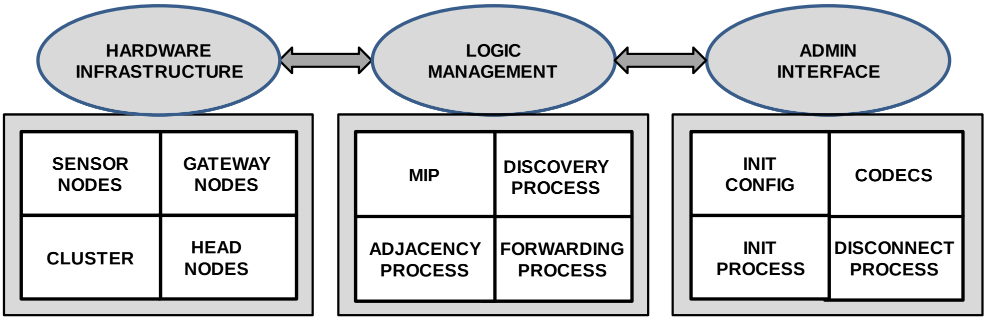

Figure 6.2. Elements of the architecture

Hardware Infrastructure level is formed by the elements in charge of building the physical and logical network topology. The physical topology is made of wireless sensor nodes. Each node can be head node, gateway node or sensor node (each node can only have one role). When a sensor node starts for its first time, it searches other sensor nodes in its coverage area. This process lets the node exchange the required information to group the nodes in clusters by means of the developed protocol. Then, the logical topology is created. Sensor nodes can only belong to a single cluster and have neighborhoods with the nodes of that cluster. The head node belongs to a single cluster but can have neighborhoods with other clusters' head nodes. The wireless connections between head clusters create a higher hierarchical level that allows exchanging information between clusters. The criteria used to determine which cluster will be the sensor node joined to, is based on the MIP associated to the sensor node, and thus, on the type of multimedia traffic it is disposed to retransmit. The sensor node will only establish neighborhoods for multimedia delivery with other sensor nodes in the WSN that are using the same MIP, so all sensor nodes in the same cluster will have the same MIP. Head nodes exchange information and control messages with other head nodes. They maintain a database with existing head nodes and clusters and the MIPs associated to them. They will deliver multimedia data to other head nodes only if the destination cluster head node has the same MIP and will retransmit the multimedia flow to the cluster nodes if the multimedia flow belongs to the same MIP. A regular node can become a cluster head when the cluster head leaves the network or fails down.

The *Logic Management* level defines the protocol elements to manage the elements of the *Hardware Infrastructure*, by using the information received from the *Admin Interface* level. MIP is the logic element that gathers the information about the multimedia streams permitted in the cluster. It is the central element of the *Logic Management* level. In this level, the logical processes (Discovery Process, Adjacency Process and Forwarding Process) that act over the sensor nodes as a function of their current state are also defined. When a sensor starts, it received the configured MIP from the *Admin Interface* level, then the Discovery Process is started and the node tries to find other nodes with the same MIP inside its coverage area. When it discovers other nodes, the Adjacency Process is started in order to create a neighborhood between both sensor nodes. These steps are followed by all new nodes in order to build the cluster. When a cluster is formed, it has the capacity to retransmit multimedia flows according with the ones defined in its MIP. Forwarding Process is started when a sensor node creates a multimedia flow request or when a multimedia flow request is received from outside of the cluster (from other cluster or from outside the WSN through the gateway). It establishes the path to follow through the cluster and reserves the resources in every node belonging to the path. It makes possible the multimedia delivery and is responsible of guaranteeing the required QoS by the MIP during the communication.

Admin Interface level allows the interaction between the user and the sensor device. There is a Graphic User Interface (GUI) that let the user modify the sensor Init Configuration, including the IP addressing and MIP selection. Admin Interface Level allows controlling manually the Init Process and Disconnect Process. The application also lets the user connect or disconnect the node to the WSN. The user can only make changes before the Init Process starts, so if a change is required, the sensor node must be stopped by using the Disconnect Process. Then, it should be initiated using the Init Process.

The number of available MIPs that can be selected by a sensor node, as well as the properties of each MIP must be defined before the system is started. Each MIP represents a different type of multimedia traffic, so the MIP should be created taking into account the network characteristics, such as the nodes density, their location, node distribution, and radio coverage, jointly with the characteristics of the multimedia flow: type of traffic (audio or video), used codec and QoS requirements. MIP definition is adapted to each particular case. For example, in a network topology with low nodes density and mixed video and audio flows, only two MIPS can be defined, one to create a cluster for audio delivery and another cluster for video delivery. But, if there is a network topology with high sensor nodes density, and only dedicated to video delivery, but using a great variety of codec, several MIPs will be defined to split the multimedia flows that use video codecs in different clusters. The MIP assigned to the sensor includes the following information: Maximum Bandwidth (MaxBW) dedicated by the sensor node for retransmitting multimedia flows, Minimum BandWidth (MinBW) required by a single multimedia flow to be processed, Maximum Delay (MaxDelay) permitted for the multimedia flow from the source to the destination, Maximum Jitter (MaxJitter) for a single multimedia flow and Maximum Hops (MaxHops) for a message in the WSN. Each MIP is identified by one byte hexadecimal code, called HCode, and an alphanumeric code, called ACode.

Figure 6.3 shows the WSN MIP-based cluster structure. We have defined two MIPs: first one for audio flow delivery, and the other for video flow delivery. Inside each cluster there could be simultaneous flow delivery with similar characteristics because they use the same MIP.

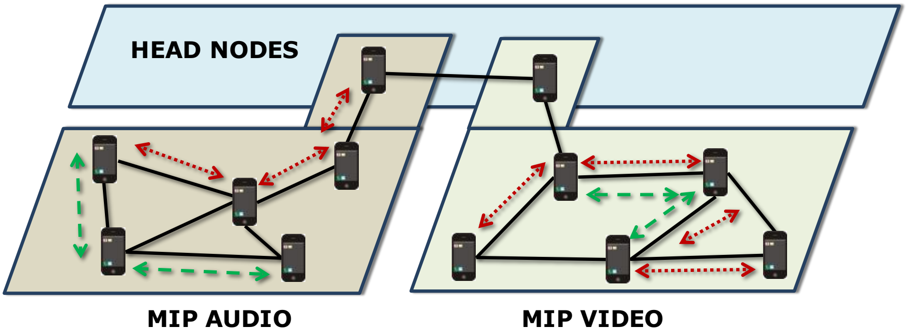

Figure 6.3. WSN structure based on MIP clusters.

6.3 Protocol Fields

The developed protocol is included in the application layer of the TCP/IP stack protocols. UDP is chosen as encapsulation protocol at the transport layer in order to reduce the processing load of the sensor node, the bandwidth consumption and the delay of the packets.

We wanted a simple protocol, with few fields, although it should be versatile. Protocol modifications should be easily made without big changes in the packet structure. Thus, we used the TLV (type-length-value) coding technique for the protocol implementation. TLV allows us to create new types of messages quickly and easily.

In Table 6.1, the protocol header fields are shown. We have included: Version, Type, Length and Value. The Type field allows us to interpret a received message. The information included in each type of message is variable and depends on the message objective, transmitter sensor node role and receiver sensor node role. Generally, the size of the message is variable, so we have defined the Length field. It provides the length of the information carried at the Value field. Using TLV coding techniques increases flexibility and scalability of the protocol and these types of messages can be extended or redefined in future revisions of the protocol very easily.

Table 6.1. Protocol Header

1 BYTE		1 BYTE	0 – 255 BYTES
2 Bits	6 Bits	8 Bits	0 – 2040 Bits
Version	Type	Length	Value

The protocol header fields are described below in greater detail:

- **Version.** This field provides the version of the protocol. Each version matches a specific and well defined messages list. All devices in the WSN must use the same protocol version to communicate properly. The size of Version field is set to two bits in order to keep reduced to the size of the protocol message. The default value of the Version field is '00 ', which matches to the protocol version 1.
- **Type.** It is a numeric code used to identify the message type. Each message Type is defined in the specific protocol version. There is a message table which includes information about message length and how the message information carried at the Value field has to be interpreted on the reception side. The size of the Type field is 6 bits, allowing a maximum of 64 message types.
- **Length.** This field indicates the length of the Value field. The numeric value is given in bytes. The Value field is variable and its size depends on the type of message. The size of the Length field is one byte, the values range goes from 0 to 255. When the value of Length field is 0 it shows that the Value field does not exist, i.e., it does not need to transmit any additional information.

- **Value.** This field holds the information to be exchanged between the sensor nodes. The size of the field can take values between 0 and 255 bytes; that matches the values of the Length field.

6.4 Data Structure

In order to carry out the required processes performed by the proposed protocol, the wireless sensor nodes have to exchange information. We have developed different types of messages with the purpose of performing next functions: Exploring the network looking for devices with similar multimedia streaming purpose, creating sensor nodes adjacencies in order to build the cluster topology, share information about the sensor nodes status, their tables and other network parameters, to start and run the multimedia flows through the cluster and notify to the neighbor nodes any event. Each defined message establishes the additional information to be included in the Value field. The following variables and structures were defined to facilitate the management of information:

- **NODE_ID**. It is the node identifier. This identifier must be unique across the whole network. The NODE_ID parameter size is 2 bytes and its value should be set before the Init Process starts, at the initialization process. There are 3 different mechanisms to generate a sensor NODE_ID: (1) Static configuration, the identifier is manually defined by the sensor administrator; (2) Automatic configuration, the last two bytes of the IP address are used as NODE_ID; and (3) dynamic configuration, where a network service uses the Multicast IP address 239.100.100.255. This last configuration option requires the previous configuration of one or more nodes as servers with pre-configured tables in order to assign the NODE_ID. This option has been only designed for testing and to facilitate the research work, but it is discouraged to use it in real environments because it introduces the need of servers.

- **NODE_RESOURCES.** This variable contains information about the available bandwidth of the sensor node for multimedia delivery. The size of this variable is two bytes. The bandwidth is measured in Kbps. The initial value of the variable is set to the MaxBW value of the assigned MIP. When a new resource reservation for multimedia delivery is made, the

NODE_RESOURCES value is decremented until the resource reservation is canceled or the delivery ends. When the NODE_RESOURCES value is below the MinBW parameter then the node changes its value to zero and no new multimedia delivery is allowed.

- **NODE_ADJ.** It is a data structure representing the connectivity state of a node into the cluster. The size of this parameter is variable and ranges between 1 and 511 bytes. First byte shows the number of adjacencies of the node in that moment. Then, the data structure is built by concatenating the NODE_ID of the neighbor node who has established a successful adjacency with. When the node starts, and it has not still been established any adjacency, the initial value of NODE_ADJ is set to 0x00 and is 1 byte in size. When the first adjacency is created the first byte is changed to 0x01 and the neighbor NODE_ID value is joined. From this point, every time a new adjacency is created, the first byte will be incremented and the new NODE_ID value will be concatenated to the NODE_ADJ structure. Because of data structure limitations, the maximum number of adjacencies by a node is limited to 255 adjacencies.
- **NODE_NCON.** It indicates the total local number of properly established and active adjacencies. The size of the variable is 1 byte. Its initial value is set to 0x00. This variable matches the value of the first byte on the NODE_ADJ data structure. NCON value is incremented or decremented each time an adjacency is created or destroyed.
- **NODE_NSEQ.** This variable represents the version number of the state table of the sensor device. The parameter size is 2 bytes. When the sensor node starts, the initialization process set its value to 0x0000. When a state change occurs, for example, when an adjacency with other node of the WSN is created or destroyed, the NODE_NSEQ value is increased o decreased. Then, the system sends a Cluster State Update (CSU) message to all nodes with successful adjacencies to update their state table. When a CSU message is received, the node compares the NODE_ID and NODE_NSEQ values on the received message with the information stored in its state table. If the value of NODE_NSEQ for this NODE_ID in the state table is below the received value, the state table is updated with the information included in the CSU message. Then, the message is forwarded to all local adjacencies except to the neighbor that sent the original CSU

message. If NODE_NSEQ of the CSU message is equal to or lower than the values of the state table, the CSU message is discarded.

- **NODE_STATE.** It is a data structure created by concatenating other local variables and structures: NODE_NSEQ, NODE_ID, NODE_RESOURCES and NODE_ADJ variables. The data structure size is calculated as a function of the number of adjacencies established by NODE_NCON value, and it ranges between 7 bytes, when there is not created any adjacency, and 517 bytes, when the maximum value of adjacencies has been reached.
- **CSU_NSEQ.** This variable is a sequence number value used in CSU messages in order to allow message fragmentation. When the size of the information in the state table cannot be fit into a single message, the CSU sequence number allows fragmenting the information into multiple messages sequentially numbered. The field size is 1 byte and the default value is set to 0x80 when no fragmentation is needed. When fragmentation is used, the packets are consecutively numbered starting from 0x01. Possible values range between 0x01 and 0xEF. When the last fragment is sent, the sequence number is increased from the previous message and, then, the first bit is changed to '1' indicating that this is the last fragment of the sequence.
- **CLUSTER_MIP.** The CLUSTER_MIP value matches the HCode value of the assigned MIP. This is a 1 Byte variable. This parameter is exchanged between neighbor nodes in the adjacency process. The MIP table is defined for the whole WSN. The number and characteristics of the defined MIP depends on the traffic pattern and multimedia flows of the network.
- **CLUSTER_N.** This parameter is used to distinguish two different clusters using the same MIP. The size of the variable is 1 byte. When the first cluster node creates the cluster and it is not aware of other clusters with the same MIP, then it selects the CLUSTER_N value equal to 0x00. Next, the first cluster node sends a request to existing cluster heads in order to know their CLUSTER_N. After this step it becomes cluster head and adds next available value from the received replies to its CLUSTER_N parameter.
- **CLUSTER_ID.** It is the cluster identifier. This value must be unique for each cluster within the same WSN. Two independent clusters into the WSN can

share the same MIP but they must always have different CLUSTER_ID value. The size of the variable is two bytes. Its value is established by the first node in the cluster. The first node is defined as the node that receives the Discovery Message ACK to establish the first cluster adjacency. The CLUSTER_ID value is built by concatenating two variables, CLUSTER_MIP and CLUSTER_N. In case of CLUSTER_ID duplications in the same WSN (because of lost messages or formed cluster joining), the oldest cluster keeps its ClUSTER_ID, and the youngest cluster changes its value to the next free value. An update message is sent to all nodes into the cluster to notify and update the new CLUSTER_ID.

- **CLUSTER_DIAMETER.** This variable shows the current cluster diameter. The cluster diameter is defined by the highest value of the lowest distance between any two nodes in the cluster. Distance between two sensor nodes is calculated by the routing algorithm. It is measured in number of hops. The size of the variable is 1 byte. When a sensor node starts, it has not established any adjacency yet, and then the CLUSTER_DIAMETER value is set to 0. Later, when the first adjacency in the cluster is created, the value is changed to 1 on both nodes. Each time a new sensor node is added to the cluster topology, the CLUSTER_DIAMETER is recalculated using the routing protocol in order to guarantee that it does not overcome the Maxhops value established in the MIP cluster. If the new adjacency exceeds Maxhops, then adjacency fails to be established.
- **MEDIA_RESOURCES**. This parameter identifies the bandwidth resources needed for a single multimedia communication. This variable is used when the sensor node creates and processes a new delivery request. Possible values can vary from MinBW to MaxBW of the assigned MIP. It depends on the characteristics of the codec used for multimedia delivery. Its value represents the bandwidth measured in Kbps and it is 2 bytes long.
- **MEDIA_SOURCE**. When a request for resource reservation takes place, the NODE_ID value of the Source Node (SN) is copied in this variable. SN is the sensor node where the multimedia delivery was originated inside the cluster. Like NODE_ID variable, the MEDIA_SOURCE size is 2 bytes. The origin of the multimedia delivery can be located outside the WSN; in this case the SN is defined as the gateway node used to enter the WSN.

- **MEDIA_TARGET.** This variable carries the NODE_ID value of the Target Node (TN). In a similar way as the SN was defined, the TN is the sensor node where the multimedia transmission ends inside the cluster. Its size is also 2 bytes. As in the previous case, the multimedia communication may finish outside the WSN, through a gateway sensor node connected to an external network. In this case, the MEDIA_TARGET is defined as the NODE_ID of the gateway node.
- **MEDIA_ROUTE**. This structure contains the full route for a multimedia packet flowing from the MEDIA_SOURCE to the MEDIA_TARGET. It is built by adding every sensor NODE_ID on the route. The route is calculated by the routing algorithm. Its size can vary from 4 bytes, when SN and TN have established a valid adjacency, to 32 bytes, when there are 16 hops on the route, the maximum number of allowed hops for any cluster. The first NODE_ID used to build the structure is the MEDIA_SOURCE and the last matches the MEDIA_TARGET.
- **MEDIA_NHOP.** This variable has the number of hops between MEDIA_TARGET MEDIA_SOURCE as it is calculated from the routing algorithm in the MEDIA_SOURCE sensor node. The size of this parameter is 1 Byte. The maximum number of hops allowed by the protocol implementation inside a single cluster of the WSN is set to 16 hops. However, the number of hops between any two nodes on a specific cluster can never be greater than the CLUSTER_DIAMETER parameter (as it is defined in the assigned MIP).
- **MEDIA_NSEQ.** This is the sequence number assigned to a multimedia delivery for the Source Node. The size of the variable is 2 bytes. The initial value is set to the hexadecimal value 0x0000. Each time a new request for multimedia delivery is originated in a sensor node the MEDIA_NSEQ value is incremented by one. This variable allows the protocol to differentiate between several multimedia flows being delivered simultaneously from the same Source Node.
- **MEDIA_INFO.** This is a data structure that contains the whole information for a single multimedia delivery that is needed and used by the remaining cluster sensor nodes. It is built on the SN when a new multimedia request is originated. The following parameters and structures are added in order to build the MEDIA_INFO structure: MEDIA_RESOURCES + MEDIA_NSEQ + MEDIA_NHOP + MEDIA_ROUTE. The size of the data structure depends on

the number of hops on the route indicated by the MEDIA_NHOP. The size can vary between 11 and 39 bytes.

6.5 Message Table

The messages used by the protocol are described in this section. Here we define the version 1 of the protocol. The TLV coding used by the protocol encapsulation allows us to change the list of messages in the following versions. For a better understanding, the whole list of messages has been organized considering the system process they belong to. UDP protocol is selected at the transport layer. Despite of this, relevant messages need to be confirmed. E.g. “ACK Discovery” is a confirmation message for the “Discovery” message and “Confirm Join” message confirms the “Request Join”.

Table 6.2 shows the protocol messages used at the adjacency process. Messages belonging to the Adjacency process are shown on Table 6.3. Table 6.4 describes the Forwarding Process messages and the Disconnect Process messages are listed on Table 6.5. System processes are detailed in the next section.

Table 6.2. Discovery Process Messages

DISCOVERY PROCESS				
MESSAGE	**TYPE**	**LENGTH**	**VALUE**	**DESCRIPTION**
DISCOVERY	0x01	3 Bytes	NODE_ID CLUSTER_MIP	Message looking for neighbors to join or to create a cluster
ACK DISCOVERY	0x02	6 Bytes	NODE_ID NODE_NCON CLUSTER_ID CLUSTER_DIAMETER	Confirms the CLUSTER_ID available to join
REQUEST JOIN	0x03	6 Bytes	NODE_ID NODE_RESOURCES CLUSTER_ID	Sensor node sends a request to built a new adjacency
CONFIRM JOIN	0x04	4 Bytes	NODE_ID CLUSTER_ID	Sensor node confirms the request to join

Table 6.3. Adjacency Process Messages

ADJACENCY PROCESS				
MESSAGE	**TYPE**	**LENGTH**	**VALUE**	**DESCRIPTION**
CLUSTER STATE UPDATE (CSU)	0x11	10-255 Bytes	CSU_NSEQ NODE_ID NODE_STATE NODE_STATE …. NODE_STATE	To exchange the State Table information between adjacency nodes. When State Table information exceeds the 255 bytes limit, it is fragmented in two or more messages.
ACK CLUSTER STATE UPDATE (ACK CSU)	0x12	3 Bytes	CSU_NSEQ NODE_ID	Confirmation message of a CSU message
NODE STATE UPDATE (NSU)	0x13	8-255 Bytes	CSU_NSEQ NODE_STATE	Information about the state of a single node. If the NODE_STATE variable exceeds 255 bytes information is fragmented in two or more messages.
ACK NODE STATE UPDATE (ACK NSU)	0x14	3 Bytes	CSU_NSEQ NODE_ID	Confirmation message of a NSU message
CLUSTER JOIN	0x15	4 Bytes	NODE_ID CLUSTER_ID	The node has filled its state table with the cluster information and it requests to join the cluster
ACK CLUSTER JOIN	0x16	4 Bytes	NODE_ID CLUSTER_ID	Confirmation message of a Cluster Join message

Table 6.4. Forwarding Process Messages

FORWARDING PROCESS				
MESSAGE	**TYPE**	**LENGTH**	**VALUE**	**DESCRIPTION**
REQUEST FORWARDING	0x21	13 – 41 Bytes	NODE_ID MEDIA_INFO	A new multimedia flow requests a resource reservation. This message is sent from the Source Node to the Target Node
RESERVE RESOURCES	0x22	13 – 41 Bytes	NODE_ID MEDIA_INFO	Target node sends a resource confirmation to the Source Node
CONFIRM RESERVE RESOURCES	0x23	13 – 41 Bytes	NODE_ID MEDIA_INFO	Reservation confirmation from the Source Node to the Target Node
QUEUE RESERVE	0x24	13 – 41 Bytes	NODE_ID MEDIA_INFO	Notification message when multimedia flow is placed in queue
REJECT RESERVE	0x25	13 – 41 Bytes	NODE_ID MEDIA_INFO	Reservation cancelation
END TRANSMISION	0x26	13 – 41 Bytes	NODE_ID MEDIA_INFO	Multimedia delivery is finished and resources have to be released

Table 6.5. Disconnect Process Messages

DISCONNECT PROCESS				
MESSAGE	**TYPE**	**LENGTH**	**VALUE**	**DESCRIPTION**
DISCONNECT	0x31	2 Bytes	NODE_ID	A node is breaking an adjacency
ACK DISCONNECT	0x32	2 Bytes	NODE_ID	Confirmation of the Disconnect message

6.6 System Operation

This section details the protocol operation. There are four main processes: Discovery, adjacency, forwarding and disconnect.

6.6.1 Discovery Process

Figure 6.4 shows the messages exchanged in the Discovery process. A sensor node starts the Discovery process when the sensor node initialization process has finished. This sensor node is named New Node (NN). The NN changes to the Discovering State and it begins sending messages looking for other sensor nodes. A "Discovery" message is sent every 60 seconds. If there are not answers after three messages, the sensor node stops sending "Discovery" messages. The Value field at the "Discovery" message has the NODE_ID, to identify the CN, and the CLUSTER_MIP to inform the selected MIP. Messages are sent to the Multicast IP address "239.100.100.CLUSTER_MIP", where the last byte matches the CLUSTER_MIP parameter. Thus only sensor nodes with the same MIP, and listening to the Multicast IP address, will receive the messages. The receiver sensor nodes are called Border Cluster Node (BCN). BCN replies by sending the "ACK Discovery" message to the NN. The "ACK Discovery" messages are sent to the unicast IP address of the new sensor node and the multicast address is not used anymore. The "ACK Discovery" message has the following information: The BCN NODE_ID, the NODE_NCON that shows the amount of established adjacencies, the CLUSTER_ID to identify the cluster and the CLUSTER_DIAMETER. When the NN receives the "ACK Discovery" message, it compares the CLUSTER_DIAMETER with the MAX_HOPS parameter; if both values are equal, then the adjacency process finishes here. If two of more clusters are available, the NODE_NCON information is used by the new sensor node in order to select the most appropriate cluster to connect with. The lowest value is preferred.

The new sensor node keeps waiting at least for 60 seconds after sending the first "Discovery" message and before selecting the target cluster, thus it allows arriving on time other possible "ACK Discovery" messages from different BCNs. When a valid cluster is discovered, and it is selected, the candidate sensor node sends a "Request Join" message to the selected sensor node (or nodes if they belong to the same selected cluster). This message contains the BCN NODE_ID that it is looking to build the adjacency, the CLUSTER_ID it wants to join and the available resources in the candidate sensor node through the NODE_RESOURCES parameter. The BCN uses the information about the NN resources to update its own state table and to notify the other sensor nodes in the cluster topology. Then, it replies the NN by sending a "Reply Join" message, it changes to the Join state and the Discovery process ends. If two or more BCNs from the same cluster are discovered the sensor node will have adjacencies with all of them.

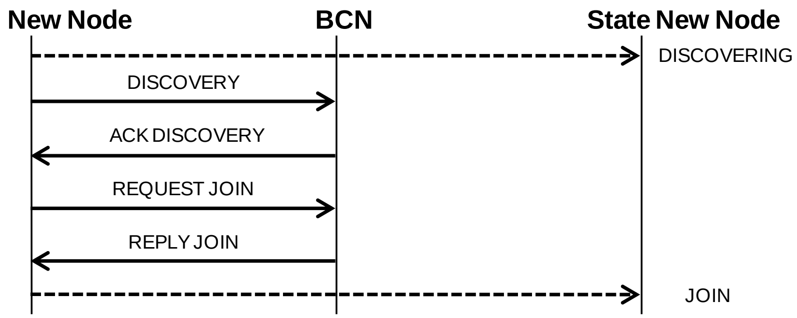

Figure 6.4. Discovery Process

6.6.2 Adjacency Process

The state table holds the information about all sensor nodes belonging to the same cluster. All sensor nodes in the same cluster share the same state table. There is a table entry for each sensor node in the cluster, thus when a new adjacency appears, the full state table is exchanged. Each table entry is stored in a single NODE_STATE structure. This structure keeps the following information about a single sensor node: NODE_ID, available resources, number of adjacencies, neighbors NODE_ID and NODE_NSEQ. If it is the first adjacency of the new sensor node, then there is only one entry on its state table and this is about its own link-status information.

The Adjacency process begins when the new sensor node makes a transition to the Join state. The exchange of messages in the Adjacency process is displayed in Figure 6.5. This image represents the specific case when the adjacency between two sensor nodes, a NN and a BCN, is successfully completed. The Inside Cluster Node (ICN) is defined as any other sensor node inside the cluster that is not going to build a direct adjacency with the new sensor node. A "Cluster State Update (CSU)" message is sent from the NN to the BCN. The message is built with the NN NODE_ID and the CSU_NSEQ. The sequence number is used to fragment the "CSU" message information when necessary. Moreover, full information on the

NN state table is included in the message; the NODE_STATE structure is used here. The "CSU" message needs always to be acknowledged by an "ACK CSU" message from the BCN; if any "ACK CSU" message is not received in 10 seconds, after the "CSU" message was sent, it is sent it again. If a sensor node sends the same "CSU" message three times, and it does not receive any answer, the adjacency process finishes unsuccessfully. After the "ACK CSU" message, the BCN sends its own state table information to the NN by sending one or more "CSU" messages.

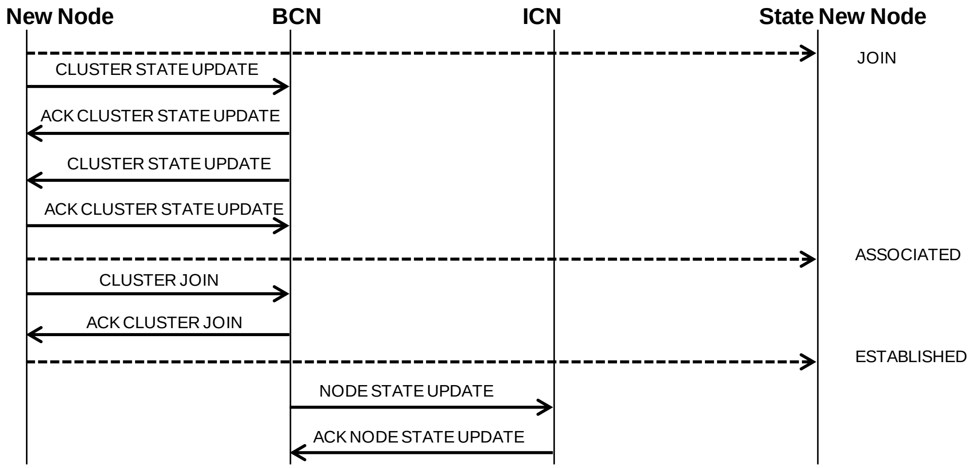

Figure 6.5. Adjacency Process

The state table is encoded in NODE_STATE structures as table entries. If the message size exceeds the limit of 255 bytes, then the information is fragmented to be sent on several "CSU" messages. The CSU_NSEQ value is used to allow fragmentation. Each "CSU" message needs to be acknowledged by an individual "CSU ACK" message in order to avoid losing information; neighbor sensor nodes need to keep the same state table, otherwise routing algorithm will not be able to calculate the most appropriated route between sensor nodes in the same cluster. After the state table has been fully exchanged between the NN and the BCN, the NN makes a transition to the Associated State. This is a transitional state, both sensor nodes are sharing the whole cluster link-state information but they have not completed their adjacency yet. BCN has not updated any other ICN yet.

When the NN changes to the Associated state it sends a "Cluster Join" message to the BCN. "Cluster Join" message has the CLUSTER_ID it is trying to join. The BCN updates its state table with the STATE_NODE information of the NN and it increases by one its NODE_NSEQ value. Then, the BCN sends a “Cluster Join ACK” message to the NN in order to accept the new adjacency. At this moment, the NN makes a transition to the Established state, which means that it has joined the cluster. The Adjacency process with the NN is completed. Finally, the NN information is flooded to the rest of the sensor nodes in the cluster by sending two “NSU” messages to all sensor nodes in the cluster. The main difference between “CSU” and “NSU” messages is that “CSU” message carries the full state table information, but the “NSU” message only carries an individual table entry for a single sensor node. In this case, two “NSU” messages should to be sent: One for the NN information and other for the BCN updated information. Every ICN checks the NODE_NSEQ for each message, then, it updates it state table and, finally, forwards the “NSU” message to all its neighbors, except to the one it has received the message from. If the NODE_NSEQ in the received STATE_NODE structure is equal or greater than the NODE_NSEQ in the ICN state table “NSU” message is ignored. Each “NSU” message needs to be acknowledged by an “ACK NSU” message, even when the “NSU” message is ignored.

6.6.3 Forwarding Process

The Forwarding process starts when there is a request for multimedia delivery in the cluster. Figure 6.6 shows the message flow diagram for the Forwarding process. The example detailed in the figure explains how a new multimedia delivery is requesting a resource reservation. The request is queued by a starved node without enough resources and finally it is processed when resources are released at the queued sensor node. Source node (SN) is defined as the first sensor node in the cluster where the multimedia request takes places. SN can be a Gateway Node, if the request is generated outside the WSN, or it can be any other sensor node if the request is generated inside the WSN. Target node (TN) is the destination multimedia flow inside the WSN; it can be a Gateway Node if the IP address destination is outside the WSN. In the diagram, the First Hop Node (FHN) has been defined as the first cluster node on the path to the Target node. FHN is calculated by the routing algorithm starting from the SN neighbors. In this case, ICN will be those nodes on the path between the SN and the TN.

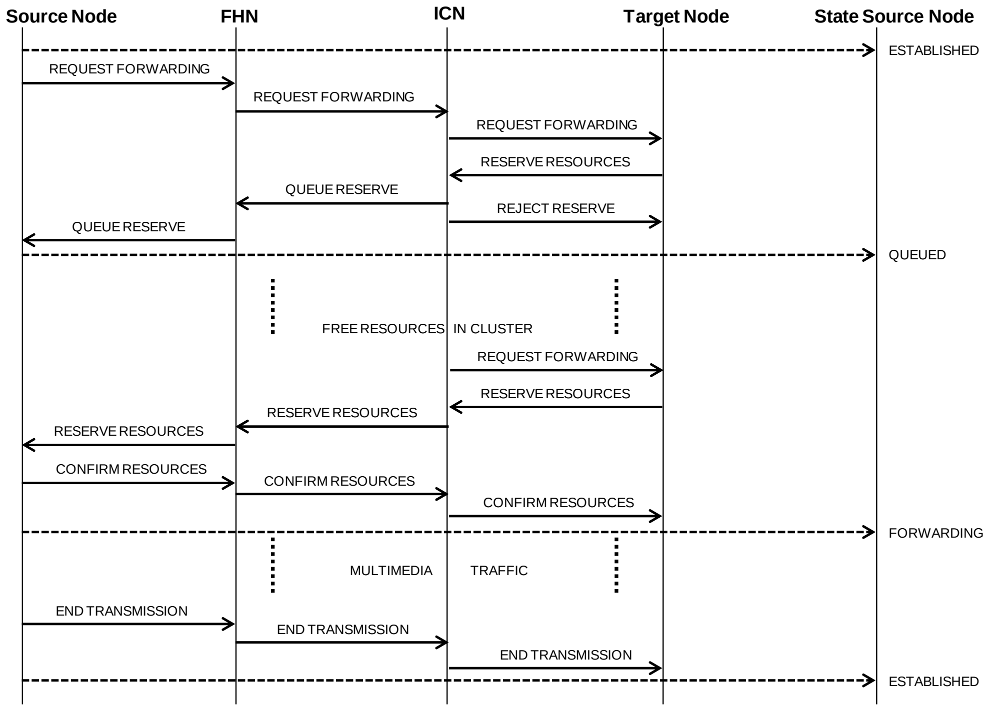

Figure 6.6. Forwarding Process

When the Forwarding process starts, the SN is in the Established state or in the Forwarding state and it receives a new multimedia flow request. First, it checks if there are enough local resources to process it. If the SN has enough available resources, the full path to the TN is calculated. The routing algorithm is used only once and only at the SN; the SN state table information contains the whole information about the cluster needed to establish each hop on the path, thus the path cannot be modified.

The message exchange starts when a SN sends a "Request Forwarding" message to the First Hop Node (FHN), which is the first NODE_ID on the calculated path. The message holds the SN NODE_ID and the MEDIA_INFO data structure. The MEDIA_INFO structure provides complete information about the multimedia request: MEDIA_RESOURCES, MEDIA_NSEQ, MEDIA_NHOP and MEDIA_ROUTE. The ROUTE_MEDIA structure contains the NODE_ID from all hops on the path, from the SN to the TN. MEDIA_RESOURCES show the bandwidth resources required to able to process the multimedia communication. MEDIA_NHOP matches the amount of hops on the path. MEDIA_NSEQ is the sequence number assigned by the SN to identify this particular multimedia flow.

The FHN receives the “Request Forwarding” message and checks if its NODE_ID is included in the MEDIA_ROUTE structure. If not, the message is discarded. If it is inside, the FHN checks it resources availability and the value is compared to the MEDIA_RESOURCES value. If there are enough local resources, the FHN reads the next NODE_ID on the hop list and the “Request Forwarding” message is forwarded to it. Resource reservation at the FHN is not set yet and can be used by current traffic, but the bandwidth resources of this request will not be used by other Request Reservation until the reservation is confirmed o rejected. All hops on the path perform the same process, hop by hop, in order to reach the TN. Finally, the TN receives the “Forwarding Request” message. The TN NODE_ID is compared with the last hop on the list provided by the MEDIA_ROUTE structure in order to check that the TN is included in this multimedia request. Available resources are checked as the other sensor nodes on the path. If there are enough resources, a reservation is made. The NODE_RESOURCES variable is decremented in the amount indicated by the MEDIA_RESOURCES parameter. This is a temporary reservation and it needs to be confirmed by the SN. Thus, the TN sends a “Reserve Resources” message back to the SN. This message also carries the MEDIA_INFO structure and it should follow the same path that the “Forwarding Request” message, but in the opposite direction. Each sensor node on the path performs a temporary reservation and follows the message back to reach the SN.

If a sensor node on the path cannot make the reservation because there is not enough bandwidth available to guarantee the multimedia communication, the designed protocol can put the request in queue for this sensor node. This process is shown in Figure 6.6, where the ICN decreases its bandwidth when it receives the “Reserve Resources” message. When an ICN is congested it can perform three different actions: It stores the request in a waiting queue, then it sends a “Reject Resources” message to the TN and finally, it sends a “Queue Reserve” message to the SN. Both messages use the MEDIA_ROUTE information in order to repeat the same path and to inform all sensor nodes in the path. The temporary resources reservation made in the sensor nodes between the ICN and the TN are cancelled by the “Reject Resources” message. On the other side, the “Queue Reserve” message releases the pre-reservation made at the sensor nodes between the SN and the ICN. The SN puts the multimedia request in a request queue and it waits to receive a notification from the congested sensor node when requested resources will be available. Both, “Reject Resources” and “Queue Reserve” messages have the NODE_ID field with the NODE_ID of the congested sensor node, ICN, thus all sensor nodes on the path can locate the congestion problems. Routing algorithm will not include congested nodes in the path. There are two options when the SN receives the “Queued Reserve” message: (1) It can wait for released resources in congested nodes, or (2) it can calculate again the path to the TN, but avoiding the congested sensor node. In this second case, the SN sends a “Reject Reserve” message to the congested sensor node; the waiting queue in the congested sensor node will be deleted. If the SN keeps the request in queue, then a timer is started in order to prevent a blocked multimedia communication. When the timer expires, a “Reject Reserve” message is sent to the congested sensor node.

If the congested sensor node keeps the request queued, it waits till other multimedia communications ends in order to have enough bandwidth resources. The forwarding process is started again from this point. Figure 6.6 shows the exchanged messages. Congested sensor node sends a “Request Forwarding” message to the TN. The original MEDIA_INFO is used. Then, a “Reserve Resources” message is sent back to the SN again from the TN. Finally, since all sensor nodes in this example have enough available resources to make the reservation, the “Reserve Resources” message reaches the SN. SN knows that all sensor nodes on the path to the TN have enough resources and they have made a temporary reservation to process the request. Then, SN sends a “Confirm Resources” message to the TN through the MEDIA_ROUTE and temporary reservations are confirmed. The SN changes to the Forwarding state and the multimedia delivery begins.

When the multimedia delivery ends, SN sends an “End Transmission” message. This message carries the MEDIA_INFO structure, which is sent to the TN to inform each sensor node that the delivery has finished and the allocated resources can be released.

6.6.4 Disconnect Process

Figure 6.7 shows the exchanged messages in the Disconnect process. Disconnect process is started by the sensor node to shut down or reboot. The sensor node sends a “Disconnect” message to each neighbor. Then, a 10 seconds timer is activated waiting the “ACK Disconnect” message. If no neighbor sends the “ACK

Disconnect” message in the timer interval, then the “Disconnect” message is sent again until 3 times. After it, the sensor node leaves the cluster.

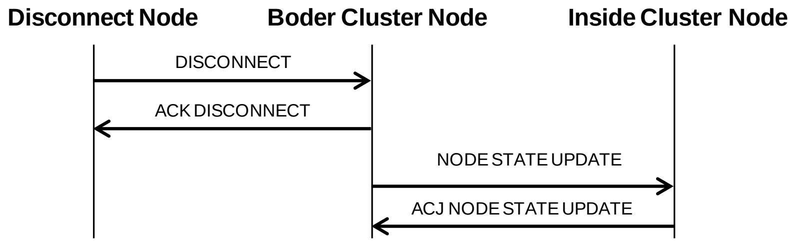

Figure 6.7. Disconnect Process

Next, neighbor sensor nodes update their status table. All infomation about the disconnected sensor node is removed. Then, they send a “NSU” message to their neighbors. The “NSU” message is flooded across the cluster, in order to let all sensor nodes update their state table. Every “NSU” message is acknowledged by the “ACK NSU” message.

6.7 Conclusion

In this chapter, we have designed and developed a new communication protocol that creates ad hoc clusters based on the multimedia flow features that are delivered inside the WSN. In order to achieve this goal, we have defined the MIP as a logical scheme that let us manage the QoS requierements and the features of the sensor nodes building the cluster. The protocol allows the creation of clusters with a maximum diameter, which is adequate for each type of multimedia flow, and selects the most appropriate nodes, with enough resources, to be in the path of the multimedia delivery. We have detailed the protocol features, the designed messages and the used variables. Moreover, we have explained the processes of the architecture, detailing how neghbour dicovery, neighborhood creation and multimedia delivery are taken place.

The content included in this chapter has been published in an international journal with ISI Thomson Impact Factor. It has been listed in the section “publications derived from the PhD”, in the conclusion chapter.

Chapter 7. Performance Tests and System Validation

7.1 Introduction

This Chapter has been split in 4 sections. The first section includes the performance test of the multimedia wireless ad hoc network, which was presented in section 4.2. The second section shows the performance study of MWAHCA, which was detailed in 4.3. The third section presents the performance study of the Fault Tolerance of MWAHCA. Finally, the fourth section shows performance Study of the QoS-Based Wireless Multimedia Sensor Cluster Protocol.

7.2 Performance test of the multimedia wireless ad hoc network

7.2.1 Test Bench and measurement system

In order to validate our multimedia-oriented protocol we deployed a real ad hoc network with 20 devices. Because we wanted to avoid any dependence with the devices characteristics, we used the same hardware configuration (Intel ® Core ™ 2 Quad CPU @ 2.50 GHz with 2 GB RAM). These devices were connected through a wireless interface, which used IEEE 802.11g standard. The wireless channel used to perform our test bench was 2.412 MHz. As a routing protocol, we used the OLSR protocol to route the information inside our ad-hoc network, because it provides optimal routes (in terms of number of hops) and we have proven that it is the best in delay at application layer.

The multimedia services offered in the ad hoc network were audio and video streaming. The whole network had an estimated diameter of 9 hops and the 20 nodes were distributed around an area of two hundred square meters.

Our test bench consists on showing the delay, jitter, packet loss and bandwidth for a node when the same multimedia service is available by several nodes at different hops. Next subsection compares aforementioned QoS parameters when the same service is available at 1, 2, 3 and 4 hops. The same case may also happen when suddenly, the multimedia service becomes unavailable when an intermediate node fails down or leaves the network, and it becomes available at different hops. Figure 7.1 shows the topology. The requester service node was node 16 and the multimedia service was available in nodes 15, 12, 8 and 2.

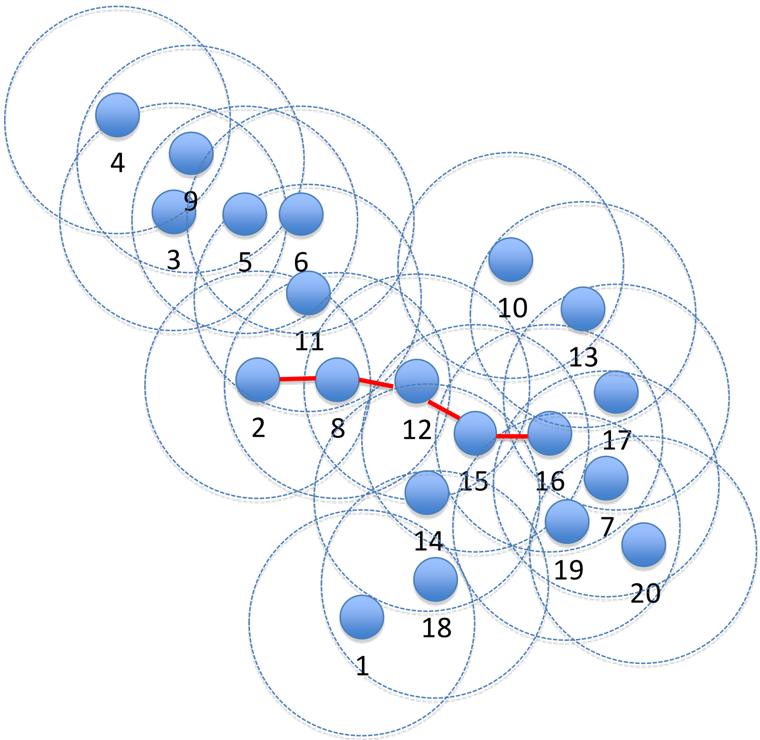

Figure 7.1. Test bench topology

In order to test the proposed protocol for different quality of multimedia streams, several audio and video codec had been selected. Codec were chosen according to their relevance in the real world. Next subsections will show the performance study for audio and video delivery in the wireless ad hoc network during 90 seconds (this time allows us to see different behaviours in detail).

7.2.2 VoIP test

We chose G.711 and G.729 codecs for VoIP streaming. The first one is one of the oldest codec used in digital telephony and it is widely used in VoIP communications nowadays. The second one is a fast codec that is becoming more and more popular because of its voice quality and high compression.

In a G.711 audio communication channel, an analog audio stream of 4 KHz is sampled at 8.000 samples per second and, then, it is codified to deliver a digital stream of 64 Kbps. Next, audio streaming is fragmented in fifty packets per second. Each packet transports 20 milliseconds of voice signal. The encapsulation cost for every packet (adding RTP, UDP and IP headers), makes G.711 to consume over 80 Kbps. Audio traffic pattern is always constant as a voice packet is transmitted exactly every 20 milliseconds.

Figure 7.2 shows the delay (in ms.) of an audio streaming communication using G.711, when there are 1, 2, 3 and 4 hops. We can see that there is a delay close to zero in all cases except in 3 hops, which means that suddenly there may be peaks of delay when there are 3 hops or more. We have not seen such delay in lower number of hops.

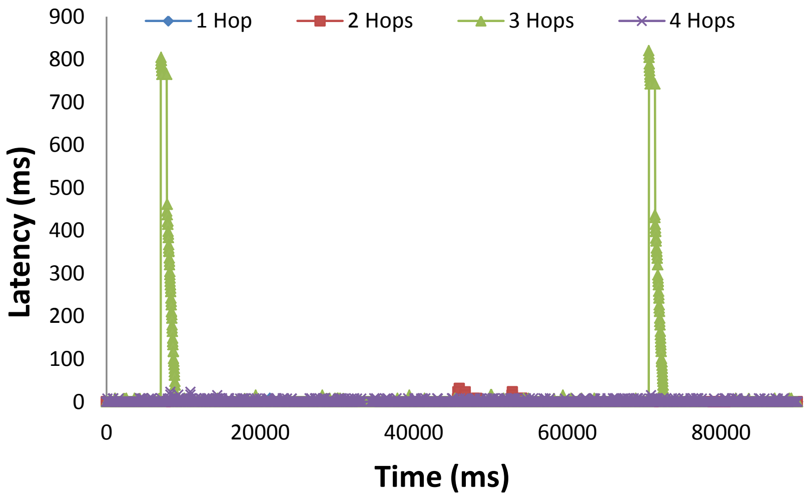

Figure 7.2. Delay of G711 for 1, 2, 3 and 4 hops

Figure 7.3 shows the jitter (in ms.) when we used G.711 in the audio streaming communication. There were two jitter peaks which happen at the same time of the peaks in the delay graph. The jitter average has been below of 1 ms. in all cases. The highest average values were obtained for 3 hops.

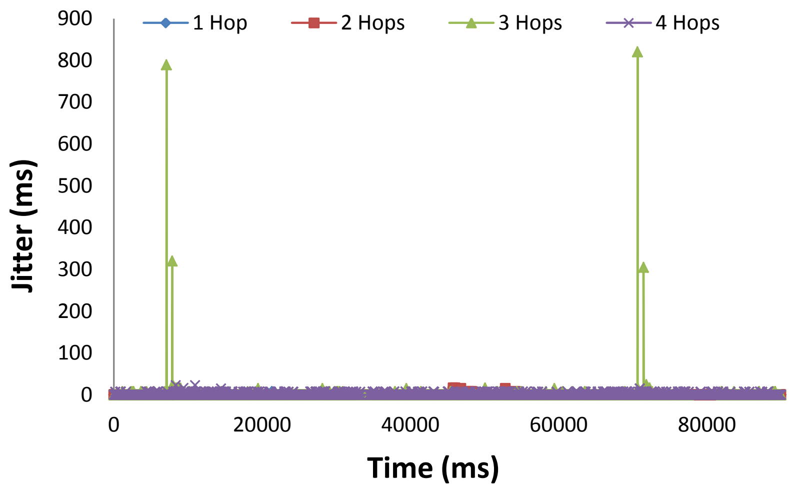

Figure 7.3. Jitter of G711 for 1, 2, 3 and 4 hops

When we measured the number of Lost packets for 1, 2, 3 and 4 hops (Figure 7.4), we observed that the case with high amount of packet loss where 3 hops. It never happened for 1 or 2 hops. There was an average packet loss of 0.06 % approximately.

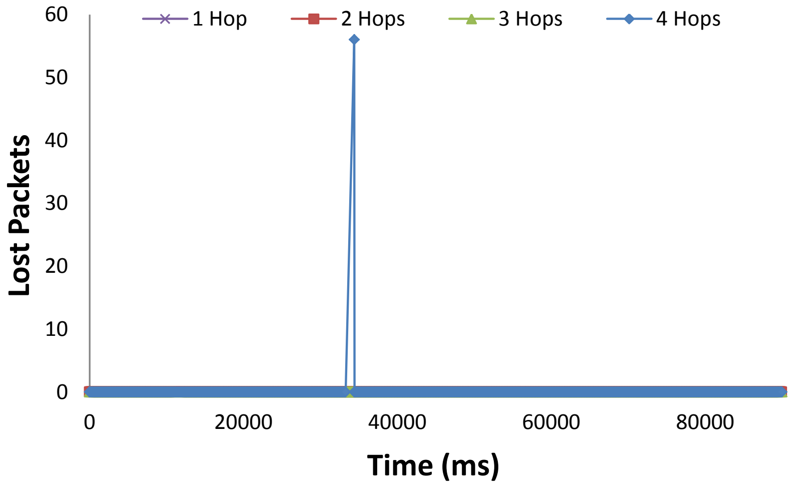

Figure 7.4. Packet Loss of G711 for 1, 2, 3 and 4 hops

In order to show the amount of bandwidth consumed in the channel in each case we measured the amount of Kbps during this audio streaming. In Figure 7.5 we can observe that the amount of bandwidth in all cases is around 89 Kbps. The case that presented more low peaks was for 3 hops. The maximum amount of bandwidth has been 94 Kbps in all cases and the lowest was given for 3 hops (3.5 Kbps).

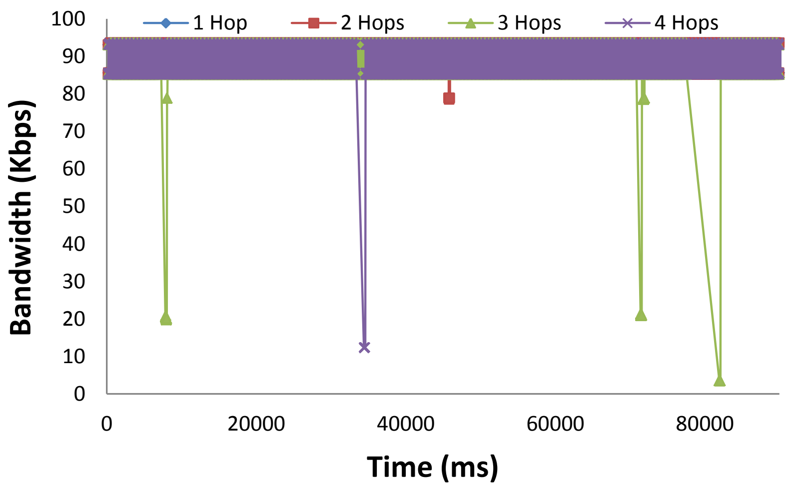

Figure 7.5. Bandwidth of G711 for 1, 2, 3 and 4 hops

G.729 codec is more efficient than G.711 codec. It reduces the payload from 64 Kbps to 8 Kbps and the final bandwidth is over 24 Kbps. So, it is possible to transmit three G.729 channels instead of one G.711 channel. For this reason, we selected G.729 codec for our second performance test.

Figure 7.6 shows the measured delay for G.729 audio communication. The case that presented more peaks was 3 hops. It also presented the highest average (3.47 ms.).

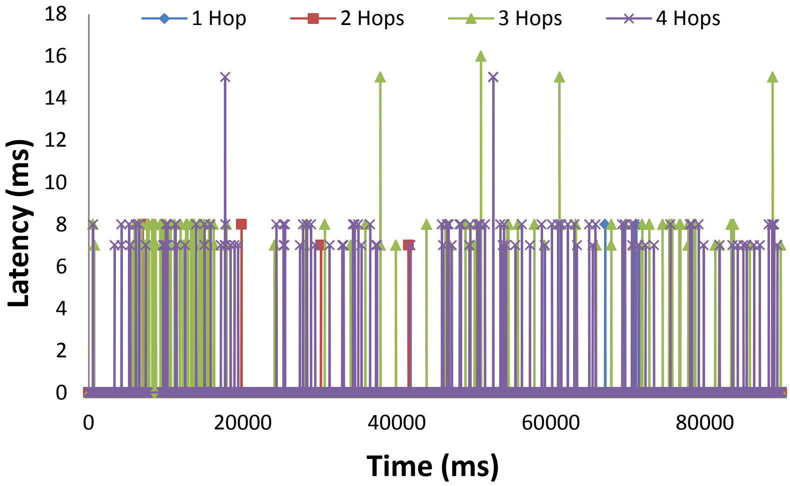

Figure 7.6. Delay of G729 for 1, 2, 3 and 4 hops

Figure 7.7 presents the jitter measured when the audio stream used G.729 codec. 3 hops was also the case with more peaks. Its maximum value was 781 ms. The minimum average values were given for 1 and 2 hops (0.01 ms).

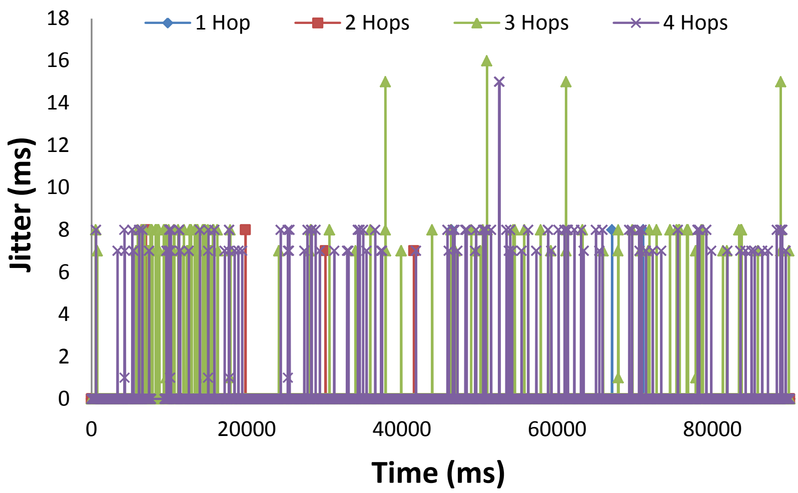

Figure 7.7. Jitter of G729 for 1, 2, 3 and 4 hops

In Figure 7.8 we can see the packet loss measured for all cases. We can see that the highest peak was given for 4 hops. Despite of this, the highest average values was obtained by 3 hops.

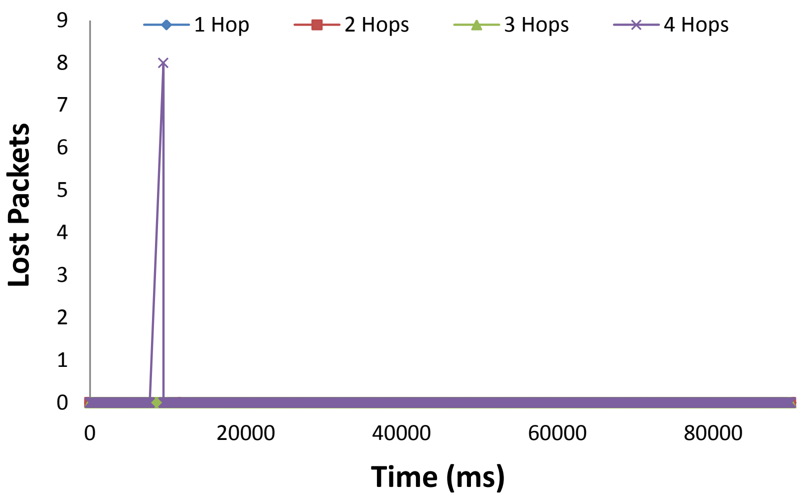

Figure 7.8. Packet Loss of G729 for 1, 2, 3 and 4 hops

In this codec the highest bandwidth values were given by 1 hop and 2 hops (as it can be seen in Figure 7.9). This means that there have been more packet loss in the other cases so less Kbps have reached the destination (we were using audio streaming, so there were no retransmission of lost packets). The lowest peak was obtained for 4 hops.

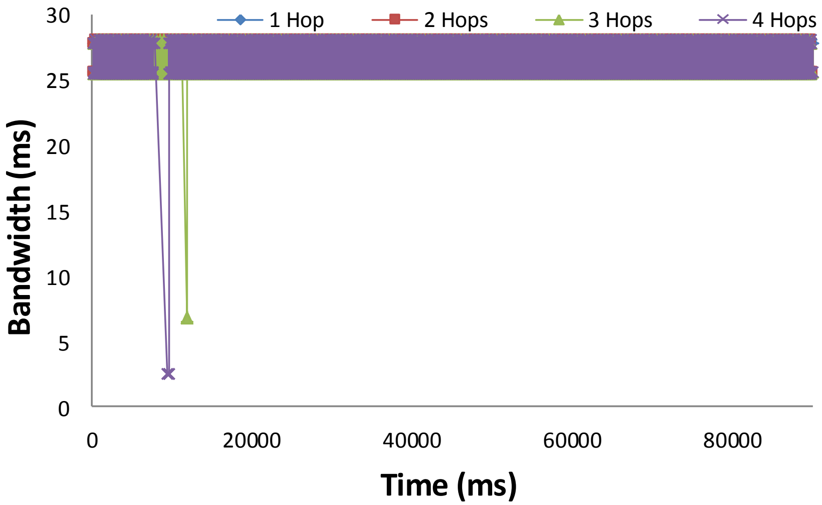

Figure 7.9. Bandwidth of G729 for 1, 2, 3 and 4 hops

7.2.3 IPTV & VoD

In order to compare the behavior of the video streaming when there are different hops from the multimedia service provider to the requester, we used a raw video that was compressed to MPEG-2 codec at different bitrates. Image compression is performed by the algorithm codecs in order to fit the output signal to the available network bandwidth, so, video streaming quality is based on codecs' selected parameters such as frames per second and screen resolution. The original video quality is 1080 x 720 pixel resolution and 30 frames per second. Output video fragments have been codified to achieve 600, 900, and 1800 Kbps bitrates. This range of values represents a typical video transmission when using different types of devices and systems such as low resolution mobiles or high quality video on demand systems for mobile devices. Higher video bitrates than 1800 Kbps will not improve the video quality seen in the mobile devices because the screen sizes of the mobiles will not let the user distinguish any difference with our highest value in this test bench.

First, we measured the delay when a user was requesting video streaming with a bitrate of 600 Kbps to nodes at different hops. Figure 7.10 shows that 1 and 2 hop cases present very low delay but in 3 and more hops, the delay is unacceptable for video streaming. 1 and 2 hops have an average delay close to 4 ms., while 3 hops reaches up to 8 seconds.

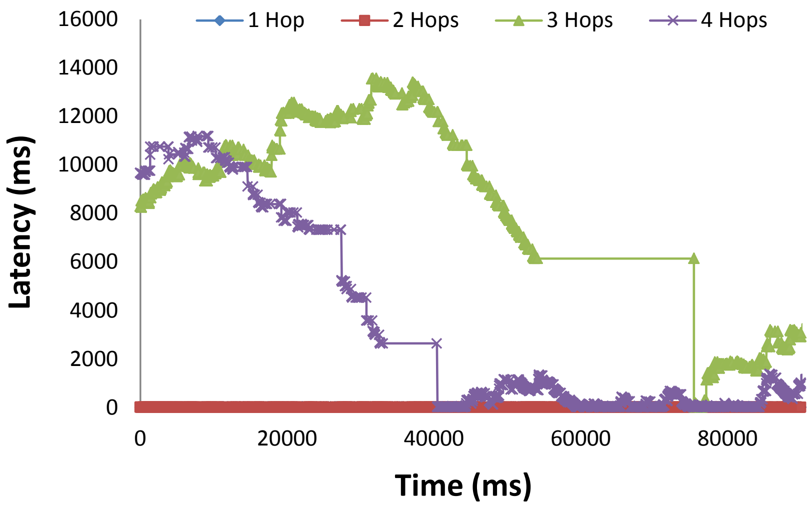

Figure 7.10. Delay of a 600 Kbps video for 1, 2, 3 and 4 hops

In Figure 7.11 we can see the jitter obtained for all cases. Although the highest peak is given for 3 hops case, the highest average has been obtained for 4 hops, while 1 and 2 hop cases remain quite similar.

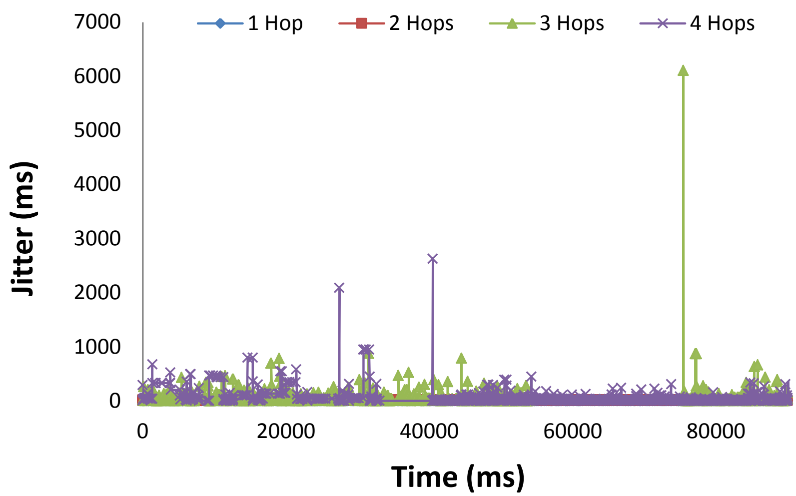

Figure 7.11. Jitter of a 600 Kbps video for 1, 2, 3 and 4 hops

Figure 7.12 shows the packet loss obtained for all cases. Although the highest peak is registered for 3 hops case, the packet loss average is about 1.69% for that case.

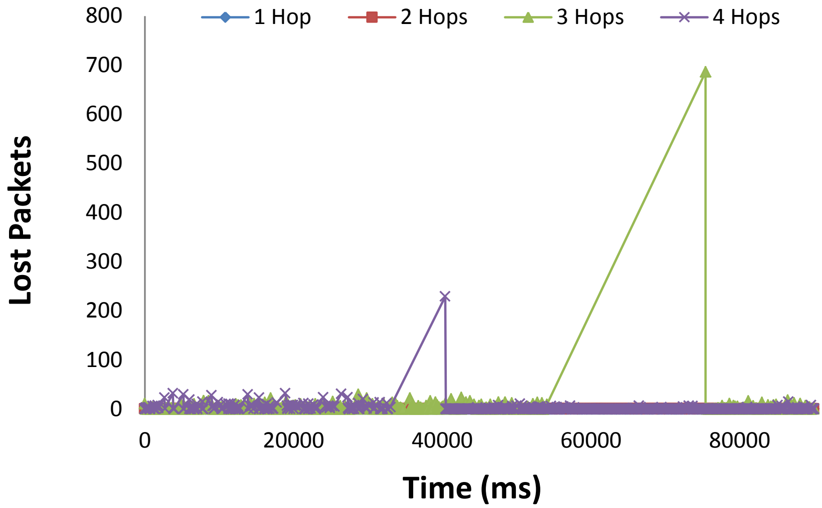

Figure 7.12. Packet loss of a 600 Kbps video for 1, 2, 3 and 4 hops

When we measured the bandwidth (Figure 7.13), we obtained approximately the same values for 1 and 2 hops. 4 hops showed an average bandwidth value that doubled 3 hops case. Again the highest peak was registered for 3 hops (5500 Kbps).

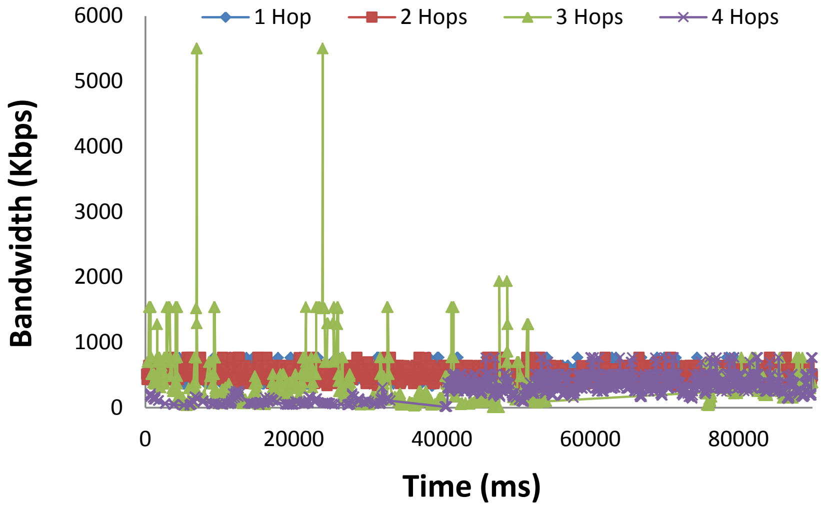

Figure 7.13. Bandwidth of a 600 Kbps video for 1, 2, 3 and 4 hops

In order to test the behaviour of our proposal for higher quality videos, we increased the bit rate of the video to be delivered through the ad hoc network. So, we tested the streaming of a 900 Kbps video.

When we measure the delay for a 900 Kbps video (figure 7.14), we observe that the only two acceptable cases are 1 hop and 2 hops. 3 and 4 hops video streaming did not finish their video delivery successfully. We have observed that 1 hop gives a delay of 5.8 ms, while 2 hops provided a delay of 26.5 ms., although the maximum delay value for 2 hops has been 2586 ms. It did not affect significantly to the video view.

Figure 7.15 presents the jitter for 900 Kbps video in all cases. The most stable behavior has been obtained for both 1 and 2 hops (with average values of 7.36 ms. and 10 ms. respectively), but 2 hops had several important peaks (the maximum peak had 1727 ms.).

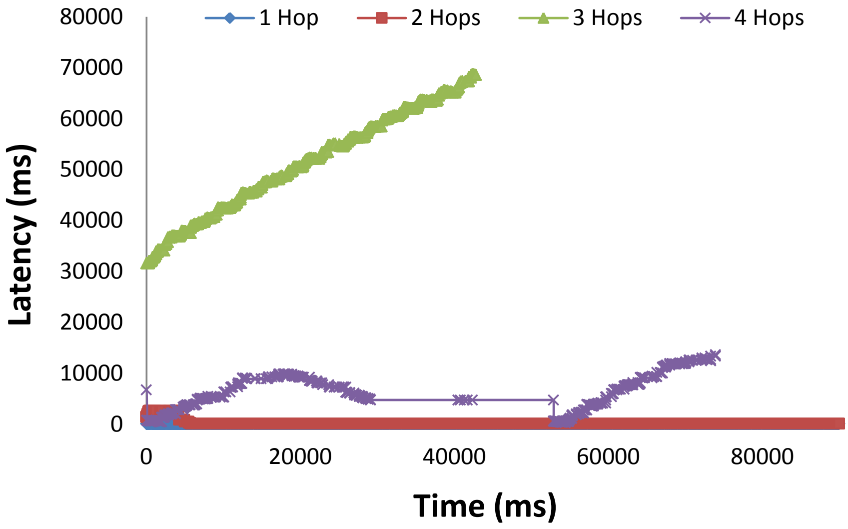

Figure 7.14. Delay of a 900 Kbps video for 1, 2, 3 and 4 hops

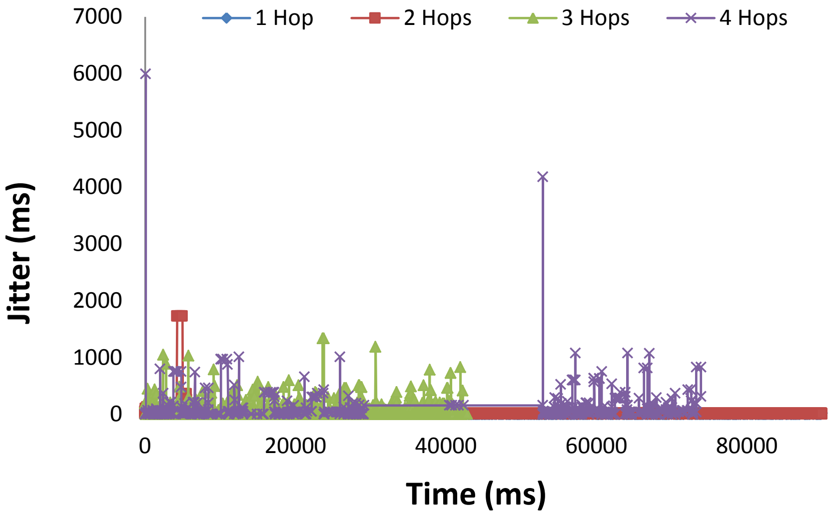

Figure 7.15. Jitter of a 900 Kbps video for 1, 2, 3 and 4 hops

Packet loss measurements are shown in Figure 7.16. It can be seen that the highest peaks were obtained for 4 hops case, which average value has been 4.22%. 1, 2 and 3 hops have obtained 0.63%, 0.05% and 0.00% packet loss respectively.

We have observed high impact in the bandwidth consumed (see figure 7.17). Because of the number of lost packets, the amount of bandwidth consumed for a 900 Kbps video in 4 hops have been 5 Kbps approximately, which means a high packet loss ratio. 1 hop and 2 hops have obtained similar values (close to 695 Kbps).

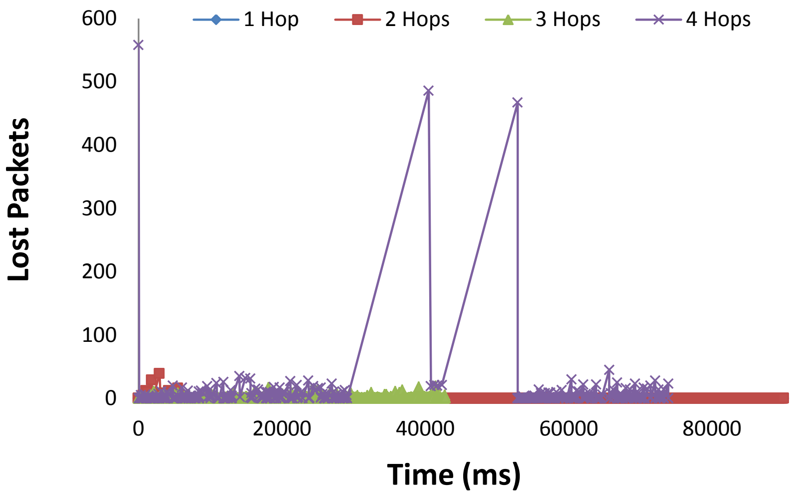

Figure 7.16. Packet Loss of a 900 Kbps video for 1, 2, 3 and 4 hops

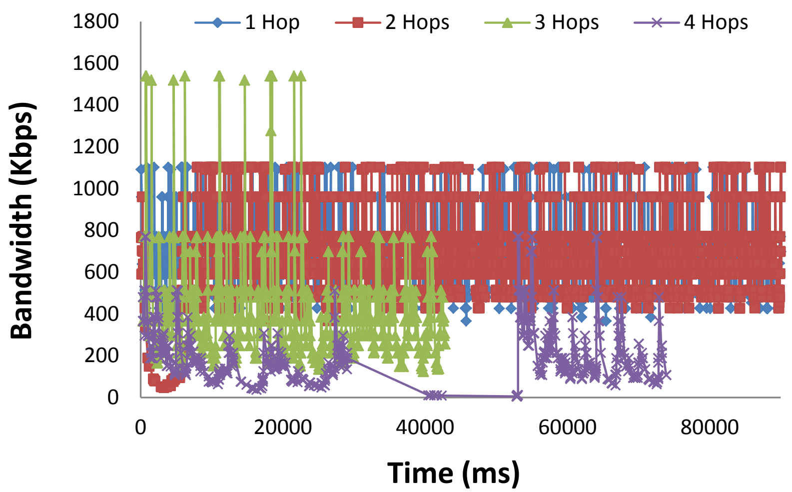

Figure 7.17. Bandwidth of a 900 Kbps video for 1, 2, 3 and 4 hops

In order to test whether our wireless ad hoc network was able to support higher bitrates or not, we configured 4 multimedia service provider nodes performing video streaming at 1800 Kbps bitrate.

Figure 7.18 shows that the delay for the 3 hops case was extremely high and the system was not able to deliver the video successfully. Although 4 hops case presented better delay, it was too high for video streaming. 1 hop and 2 hops had similar average delay (10.71 ms. and 12.74 ms. respectively). The maximum value was also very similar.

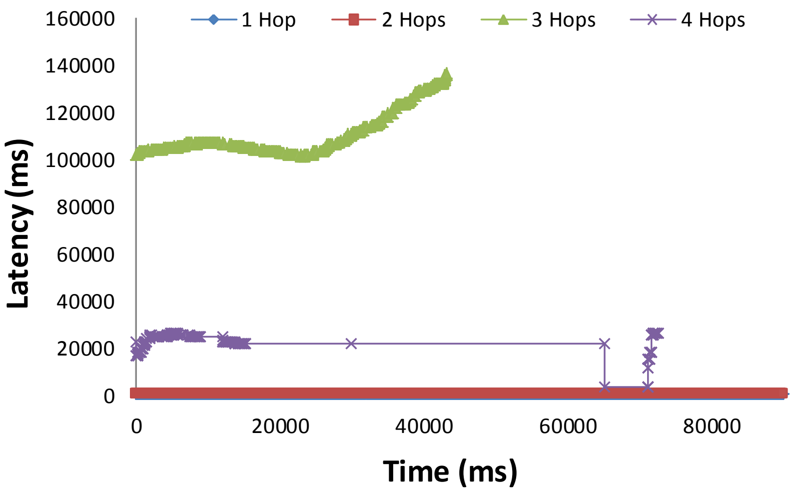

Figure 7.18. Delay of an 1800 Kbps video for 1, 2, 3 and 4 hops

In Figure 7.19, we can see that 1 hop and 2 hops present similar behavior for jitter measurements (an average value of 6.38 ms. approximately and a maximum value of 32 ms. approximately). 4 hops case presented the worst behavior.

When we analyzed the packet loss (Figure 7.20), we saw that 4 hops had an average value of 10% approximately. Again, 1 hop and 2 hops have had very few packet losses.

The bandwidth consumed in all cases for 1800 Kbps video is shown in Figure 7.21. 4 hops case has been the one with highest and lowest peaks, which means that it had a very rampant graph, but the average value has been 3 times less than 1 hop. For 1800 Kbps we have obtained an average bandwidth of 1299 Kbps for 1 hop and 1340 Kbps for 2 hops. We obtained lower bandwidth values for 3 hops than for 4 hops case.

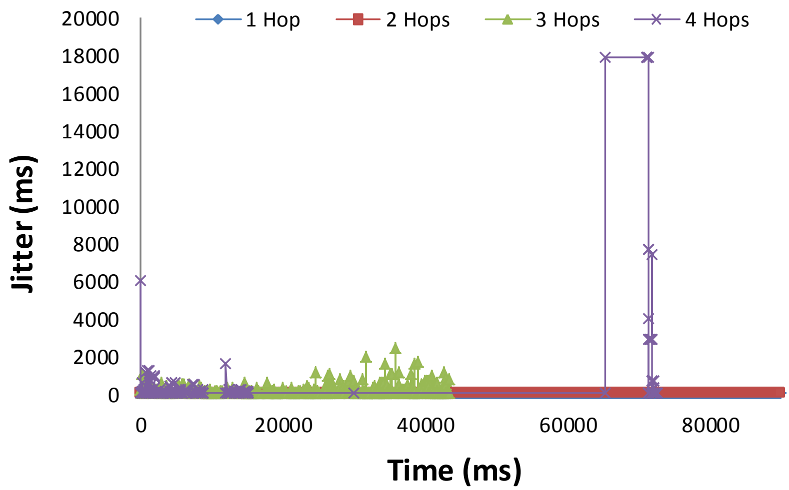

Figure 7.19. Jitter of an 1800 Kbps video for 1, 2, 3 and 4 hops

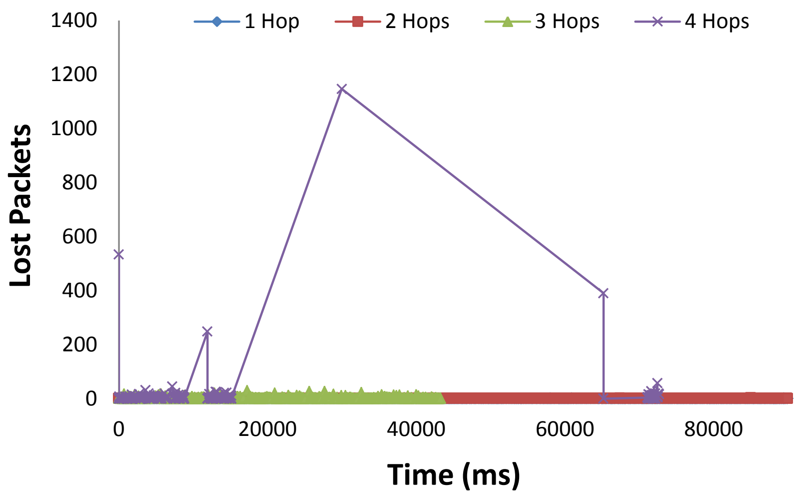

Figure 7.20. Packet Loss of a 1800 Kbps video for 1, 2, 3 and 4 hops

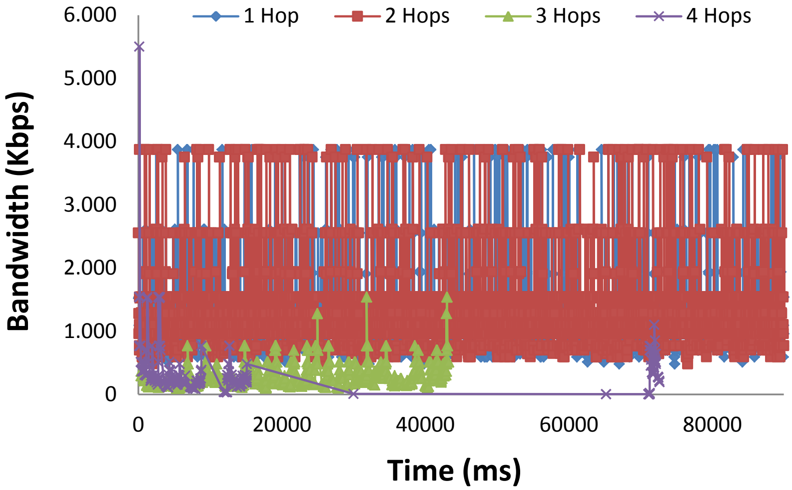

Figure 7.21. Bandwidth of an 1800 Kbps video for 1, 2, 3 and 4 hops

7.3 Performance Study of MWAHCA

When multimedia streams are sent through ad hoc wireless networks, the bandwidth and the logical topology characteristic requirements should be adjusted as a function of the type of traffic, audio or video, and the codec used for the transmission.

We have deployed our architecture with the aim to measure the delay and jitter parameters when several multimedia streams in different wireless ad hoc cluster topology configurations are set up. Obtained results will allow us to validate our protocol and architecture proposal, which groups the nodes inclusters based on the MIP. Nodes are classified and clustered based on their capacity to support different types of multimedia streams. Because we wanted to avoid any dependence with the devices characteristics, we used the same hardware configuration for all devices. The had Intel ® Core ™ 2 Quad Processor working at 2.50 GHz with 2 GB RAM. These devices were connected through a wireless interface, which used IEEE 802.11g standard. The wireless channel used to perform our test bench was 2.412 MHz.

The parameters of the cluster topology, such as the diameter, are limited based on the type of multimedia stream that is going to be used. The protocol allows several simultaneous multimedia streams guaranteeing the required resources for each one of them in their respective cluster.

We have selected the most appropriated MIPs taking into account the most used video codec characteristics. With the objective to maintain equilibrium between the flexibility of the options and maintain a reduced number of profiles, we have defined 3 MIPs for video in our test bench. The values assigned to each MIP are shown in MWAHCA description. Each node in our test bench has any of these MIPs configured before joining the network.

7.3.1 Codecs - Comparison

In order to compare the multimedia stream behaviour, we have selected three video codecs using 600 Kbps, 1800 Kbps and 3600 Kbps bandwidth consumtion. They correspond with the MIPs V2, V3 and V4 in table 4.1. We have analized their behaviour when they are being streamed over the same cluster topology and with the same experimental conditions, so the differences in the results are only caused by the codecs charactesitics used by each one of the multimedia streams. Figure 7.22 shows the delay obtained when those three video codecs are streamed during 30 seconds. In order to provide a graphical representation, we compute the average delay of the last 20 received packets, estimating the value in 100 milliseconds intervals. Being X_i the delay of a single packet, our average delay is given by expression 7.1.

$$Y_i = \frac{\sum_{j=i}^{i+20} X_i}{20} \tag{7.1}$$

La figura 7.22 provides Y as a function of the time. We can sobserve that 3600 Kbps has higher delay and has higher delay variation. 1800 Kbps and 600 Kbps are more stable. The one that provides lower values is 600 Kbps.

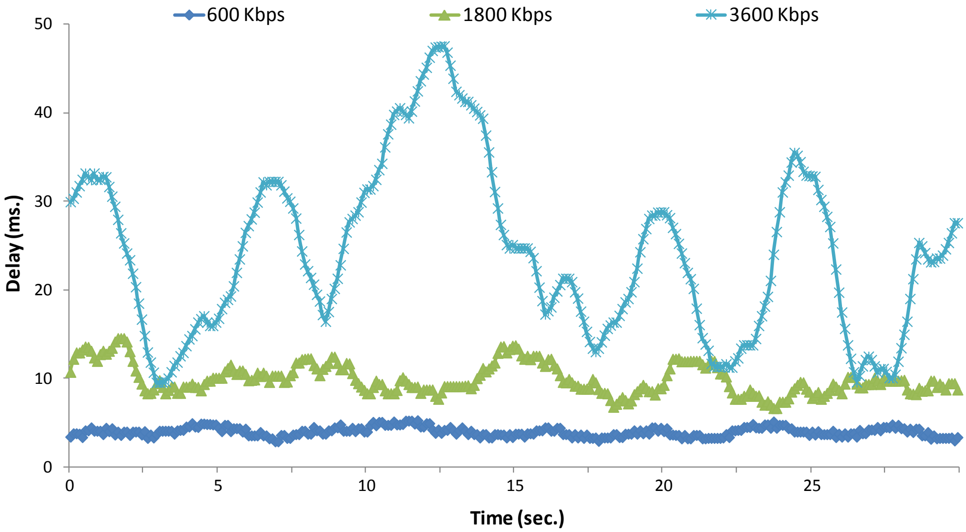

Figure 7.22. Delay of different streams using different codecs.

We have also performed an statistical analysis in order to interprete the results. In order to determine whether the observed differences in the delay are random or are caused by intrinsic characteristics of the codecs, we have defined the following null hypothesis H_0: There is not difference between the average delay obtained by the three codecs with bandwiths 600 Kbps, 1800 Kbps and 3600 Kbps. Table 7.1 shows the estimations performed for each codec. N is the number of samples, μ is the average score, σ is the standard deviation, Min is the minimum score, Max is the maximum score and Conf. Int. is the confidence interval. In order to perform the statistical analysis, we have used a confidence level (α) of 0.01, with a confidence interval of the 99%. The results show that the average delay value of each codec is outside of the confidence interval obtained for all codecs in all analyzed cases, so we can reject the null hypothesis with $p<0.01$. The highest value has been obtained for 3600 Kbps in all cases, while the lowest values has been obtained for 600 Kbps in all cases. We can conclude that the behaviour of a multimedia stream when using the same cluster topology is different and depends on the bandwidth required by the codec, so we have to use a different treatment. We have also observed that lower bandwidth consumption provides lower delay values with higher confidence.

Table 7.1. Statistical values of the delay of different streams using different codecs.

VIDEO Codecs	Parameters						
	N	μ (ms)	σ (ms)	Min (ms)	Max (ms)	Conf. Int. (ms)	
600 Kbps	300	3.96	0.51	2.86	5.23	3.88	4.03
1800 Kbps	300	9.95	1.72	6.62	14.52	9.69	10.20
3600 Kbps	300	24.27	9.57	9.52	47.58	22.84	25.70

Figure 7.23 shows the results obtained when jitter is measured as a function of the used codec during 30 seconds. Jitter values are the average jitter values of the last received samples for the three multimedia streams using the same cluster topology. The three streams use the same number of hops (2 hops). We have observed that the jitter is quite higher for the codec with higher bandwidth consumption (3600 Kbps), while it remains quite stable and considerably lower for 1800 Kbps and 600 Kbps.

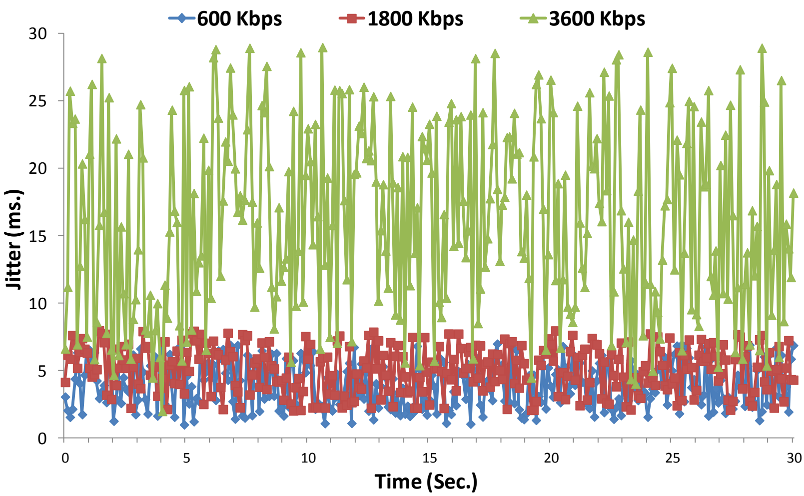

Figure 7.23. Jitter of different streams using different codecs.

The statistical analysis provided in Table 7.2 shows that there is a significant difference between the codec with 3600 Kbps and the other codecs, obtaining μ and Max (ms) values 3 times higher. There is also a significative difference with a value of α=1, between 600 Kbps and 1800 Kbps codecs.

Table 7.2. Statistical values of the jitter of different streams using different codecs.

VIDEO Codecs	Pararameters						
	N	μ (ms)	σ (ms)	Min (ms)	Max (ms)	Conf. Int. (ms)	
600 Kbps	300	3.94	2.91	1	7	3.68	4.19
1800 Kbps	300	5.11	1.72	2	8	4.85	5.37
3600 Kbps	300	16.26	48.17	2	29	6.21	17.29

7.3.2 Hops Comparison

We performed the following test with the aim to show how a multimedia stream has different quality of service values as a function of the number of hops in the wireless ad hoc cluster. In order to perform this test we have selected a codec with an average of 600 Kbps and we have tested it in four topologies with different number of hops inside the cluster. Figure 7.24 shows the obtained delay as a function of the number of hops. We have observed that 1 and 2 hops do not increase the delay much, but it is considerably increased in three hops and huguely increased in 4 hops. Delay values are not increased proportionally with the number of hops.

We have also performed an estatistical analysis based on the null hypothesis H_0: There is no difference in the delay average when a multimedia stream of 600 Kbps is being transmitted over several cluster ad hoc networks with diammeters 1, 2, 3 and 4 hops.

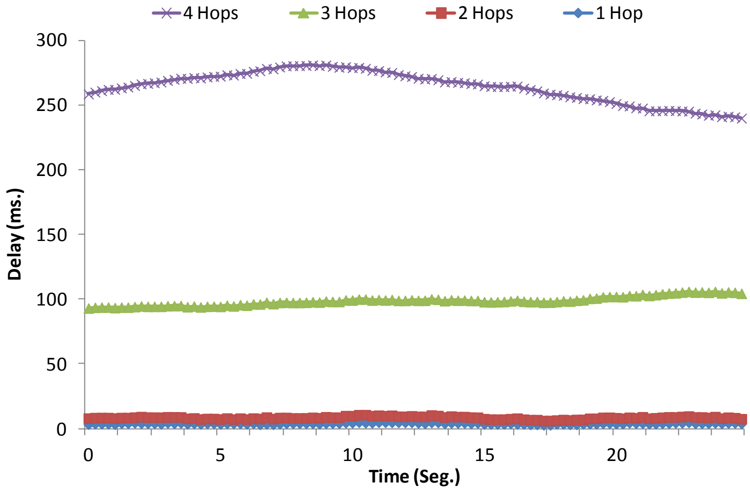

Figure 7.24. Delay of a multimedia stream of 600 kbps for different cluster diameters

Table 7.3 details tnhe estimated values for all obtained data. We have selected a confidence level (α) of 0.01, with a confidence interval of 99 %. After obtaining these resutls we can discard the null hipothesys and affirm that the delay of a multimedia stream in a cluster ad hoc topology depends in the number of hops between the Source Node and the Target Node. We have also observed that the main differente is between 2 hops and 3 hops.

Table 7.3. Statistical values of the delay as a function of the diameter

HOPS	Parameters						
	N	μ (ms)	σ (ms)	Min (ms)	Max (ms)	Conf. Int. (ms)	
1	100	3.96	0.51	2.86	5.23	3.88	4.03
2	100	4.40	0.52	3.02	6.21	4.32	4.47
3	100	96.66	5.60	85.02	106.11	95.83	97.47
4	100	165.38	72.162	158.08	1783.02	154.86	175.90

Figure 7.25 shows the measurements gathered for the jitter as a function of the number of hops in the cluster when 1800 Kbps multimedia stream is used. It shows a 30 seconds interval. We have observed that the highest values are obtained for 4 hops. The difference with the rest of cases is high. 1 hop has the lowest jitter values.

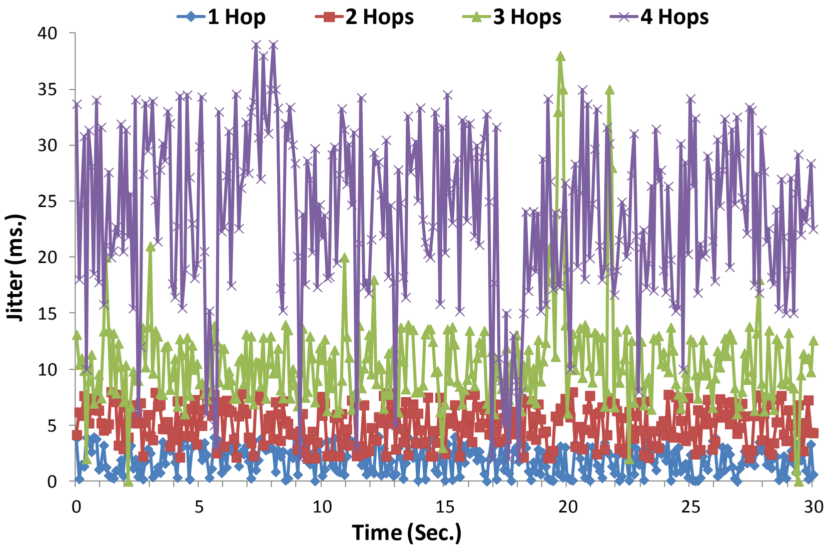

Figure 7.25. Jitter of a multimedia stream of 600 kbps for different cluster diameters

We have performed the statistical analysis of the results with α=1(see table 7.4). We can check that there is a significative difference when the number of hops is increased. 3 hops doubles 2 hops values and 4 hops doubles 3 hops values. We can conclude that the jitter values directly depend on the number of hops in the cluster topology.

Table 7.4. Statistical values of the jitter as a function of the diameter

HOPS	Parameters						
	N	μ (ms)	σ (ms)	Min (ms)	Max (ms)	Conf. Int. (ms)	
1	300	1.99	1.23	0	6	1.80	2.17
2	300	5.11	1.72	2	8	4.85	5.37
3	300	10.56	4.28	0	38	9.92	11.19
4	300	23.71	7.57	2	39	22.58	24.84

7.4 Performance Study of the Fault Tolerance

In this section, the measurements obtained from the test bench are presented. The fault tolerant mechanism described has been incorporated to MWAHCA. The protocol extension has been programmed in JAVA language. Figure 7.26 shows the topology used in these tests. Each dashed lines matches the adjacency between two different nodes. The orange node, N22, represents a congested node where the available resources have been reduced by other multimedia flows. This node has a large number of adjacencies. The paths between the gateway nodes are longer when they cross the N22 node. Black nodes are gateway nodes; GW1 is connected to the external node S1, which is running the VLC software as a video streaming server, and GW2 is connected to the external node C1, which is running the same VLC software but as a client of video streaming receiving the S1 signal. Green nodes are cluster nodes with identical characteristics; all of them have the same amount of available resources. The only criterion used by the routing algorithm is the number of hops because of the laboratory settings and the node configurations. Only a multimedia unidirectional flow is being transmitted from S1 node to C1 node. When transmission starts in S1, multimedia packets are flooded to GW1 where the developed protocol is used. GW1 extracts the multimedia information into the IP, UDP and RTP headers to build the MEDIA_INFO structure. Then, the routing algorithm calculates the optimal route to reach the GW2 node, which is the destination node into the ad hoc cluster. The N22 node is discharged because the minimum amount of hops is four while distance along the upper path, GW1 – N11 – N31 – GW2, and the lower path, GW1 – N13 – N33 – GW2, is only three hops. There is not load balance implemented, therefore the routing algorithm selects the route which next hops has a lower NODE_ID. In figure 7.26, it is N11 node. In summary, if the video streaming starts after the MIP cluster has converged, the route selected will be: GW1 – N11 – N31 – GW2, with three hops. We will call this initial route as *baseline path.*

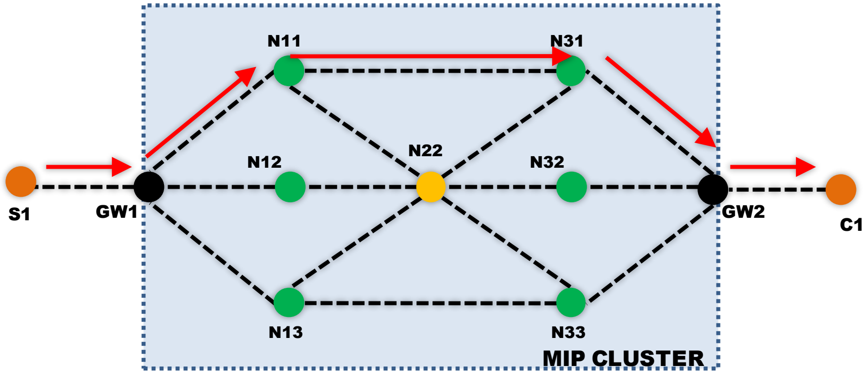

Figure 7.26. Experimental topology

Once, all nodes in the MIP cluster are working properly, we gathered measurements of latency, jitter and lost packets at C1. They represent the optimal values for this specific network arrangement because the optimal path has been established when all nodes are working and no other multimedia flow or IP traffic is on the ad hoc network. Measured values for QoS parameters depend on the bandwidth consumption spent by the selected video codec. Latency, jitter and lost packets values of a video streaming with a higher bandwidth consumption codec will also be higher. In order to validate the results, independently of the codec features, three different codec have been selected in this study: 600 Kbps, 1.500 Kbps and 3.000 Kbps.

While C1 is receiving video streaming, a node failure is introduced in the network by forcing to shutdown N31 node. Following the fault tolerant mechanism introduced in this subsection, the N11node becomes active node when the N31 failure is detected. Thus, it looks for an alternative path through the other neighbor node, N22. We called this temporary route as *fast switching path.* It consists of the optimal path from GW1 to N11 concatenated to the new temporary route calculated by the routing algorithm between N11 and GW2. The whole *fast switching path* is GW1 – N11 – N22 – N32 – GW2. The expected results are that QoS values in this stage must be worse than *baseline path* values because this is not the optimal route and no resources reservation has been performed on N22, N32 and GW2 nodes. However, if fault tolerant algorithm is working properly, the video streaming will not be interrupted at anytime.

At the final phase, the recovery mechanism tries to find a new optimal path between GW1 and GW2. When the active node, N11, sends the *fault node message* to GW1, a new forwarding process starts in the source gateway node. Looking the diagram of the topology we can see that the new optimal path will be composed by the following nodes: GW1 – N13 – N33 – GW2. This third and definitive path is called *Recovered path*. Then, measurements of latency, jitter and lost packets are performed using a video streaming of 600 Kbps, 1.500 Kbps and 3.000 Kbps in three different conditions:

- **Baseline path.** Average values are calculated from the last 100 multimedia packets before the N31 node fails.
- **Fast Switching path.** Average values are calculated from all packets received by C1 in this stage. While this is a temporary path, only a limited number of packets (usually less than one hundred) will be sent along this route.
- **Recovered path.** Average values from the first 100 multimedia packets received by the C1 node taken 10 seconds after the N31 went down. A 5 seconds waiting time is established to ensure that the MIP cluster topology has converged and the fault tolerant mechanism has already established the recovered path properly. This value has been calculated from preliminary tests.

Figure 7.27 shows the average latency values in different stages of the fault tolerant process. We can see how the average latency is similar when we compare the baseline and 10 seconds after the node failure happens, in the recovered path stage. This result demonstrates that the multimedia transmission has been recovered and the latency values in the new optimal route are similar to the values achieved before the failure. On the other hand, the average latency in the fast switching path is significantly higher than in the other two paths. This result has a logic explanation because the temporary path built in the fast switching state is longer (4 hops in the ad hoc network) that the initial and final path (3 hops). We must also consider that some packets were buffered at the active node while the routing protocol was running. Latency for individual packets in this phase ranges between 100 and 300 ms, depending on the used codec. From these results we can conclude that latency values are valid for multimedia transmission, either unidirectional or bidirectional flows, in the three experimental conditions because of the fault tolerant mechanism.

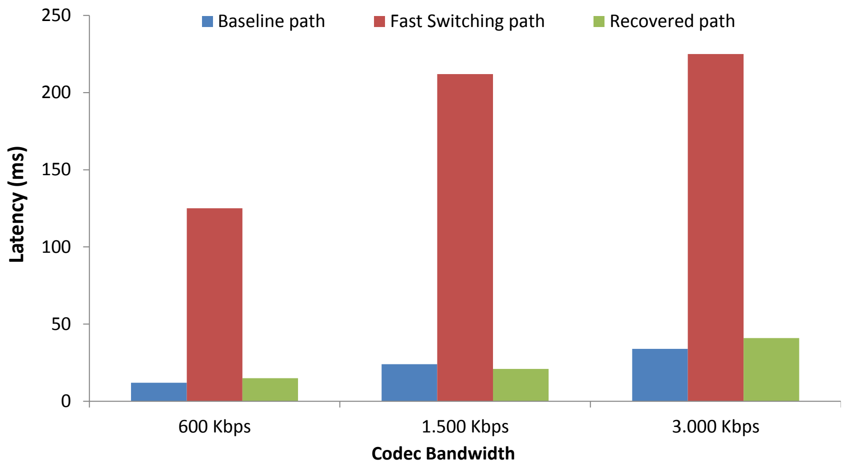

Figure 7.27. Latency in different stages of the fault tolerant process

Figure 7.28 shows the average jitter measures. As it happened with the latency, jitter in the baseline and recovered path are similar and, at the same time, they are considerably lower than in fast switching path. Nevertheless, in this case, jitter values in the temporary path have values that could be considered too much high for a real time bidirectional multimedia communication. This negative effect can be successfully eliminated in an unidirectional transmission, like a video streaming, by setting up a jitter buffer in the node at the end of the communication. For example, in the gateway destination node, just before leaving the MIP cluster or, even, at the end user device, C1. But, in a bidirectional real time transmission, like VoIP or video conference, the jitter buffer cannot take high values without harming the latency values.

Figure 7.29 shows the average lost packets percentage measured in each phase with identical criteria than in the previous cases. One more time obtained results confirm the multimedia flow has been completely restored when the fault tolerant mechanism is implemented. In this particular case, lost packets in the codec with higher bandwidth requirements, 3.000 Kbps codec, shows an excessively large number of lost packets (nearly 50 percent in the fast switching path).

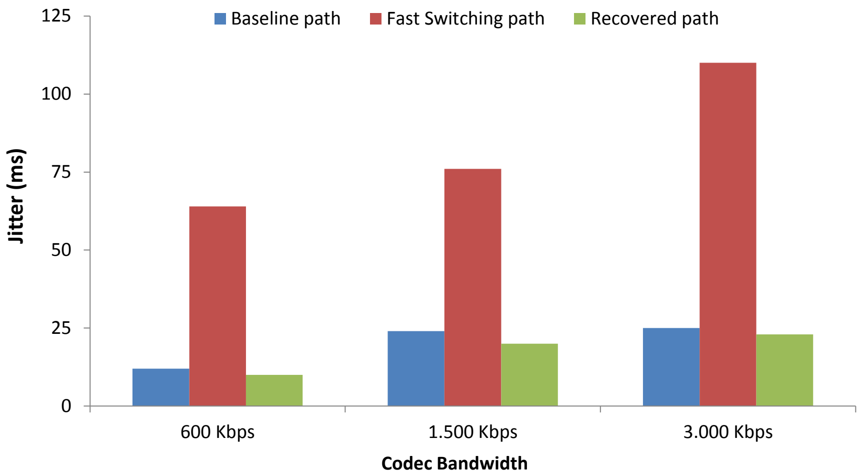

Figure 7.28. Jitter in different stages of the fault tolerant process

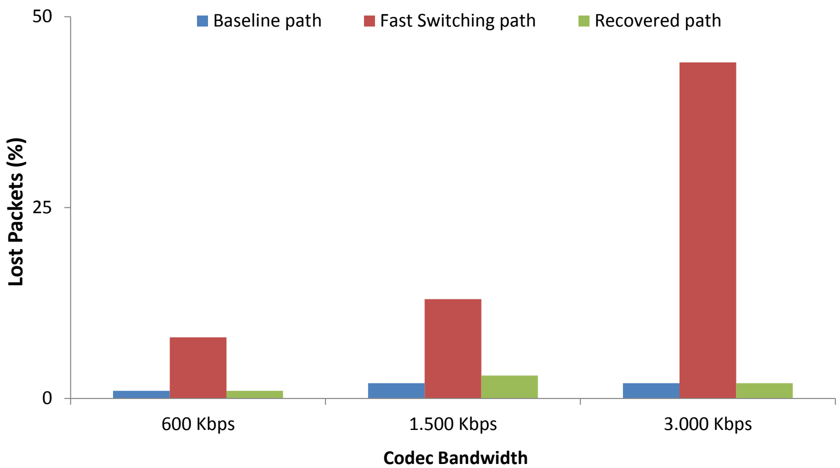

Figure 7.29. Lost Packets in different stages of the fault tolerant process

Finally, with the same test bench and the same settings, we studied the average convergence time in four different experimental conditions:

1. The proposed fault tolerant mechanism is not running and, moreover, N31 node is shutdown. This means that the original protocol is used and N31 node sends a message to N11 node before it fails down. Thus, N11 node knows the topology change before the *baseline path* is interrupted.
2. The fault tolerant mechanism is not running but, now, N31 node is suddenly shutdown. Then, the original protocol is used and N31 node does not send any message before it fails down. The *baseline path* is interrupted before N11 node knows the adjacency with N31 is not valid anymore.
3. The developed fault tolerant mechanism is running and N31 node is shutdown. Then, the fault tolerant features proposed in this subsection are incorporated to the protocol and N31 node sends a message to N11 node to break the adjacency; thus N11 becomes an active node before the *baseline path* is lost.
4. The developed fault tolerant mechanism is running and N31 node is suddenly shutdown. Then, the proposed fault tolerant features have been incorporated to the protocol and N31 node does not send any message before failing down.

Figure 7.30 shows the convergence time for each one of the above conditions. Convergence time has been calculated as the difference between the time of arrival of the last packet before lost packet happens and the first packet after the multimedia transmission is restored over an optimal and guaranteed path. Results show an evident difference between the conditions. The fault tolerance is used in conditions 3 and 4 and conditions 1 and 2 do not use the proposed algorithm. The fault tolerant mechanism achieves a convergence time lower than half a second when the proposed algorithm is used.

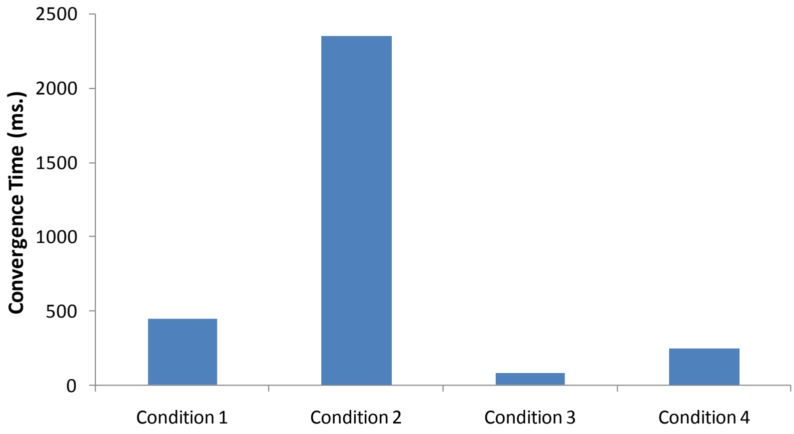

Figure 7.30. Convergence time

7.5 Performance Study of the QoS-Based Wireless Multimedia Sensor Cluster Protocol

In order to validate the proposed algorithm we have designed and built a test bench. Our protocol organizes sensor nodes in four clusters, two audio clusters and two video clusters. Each cluster has assigned a different MIP. When a NN starts in the wireless network it knows the MIP that it belongs to. Then it tries to discover other sensor nodes with the same MIP and finally it joins the cluster. If it is the first sensor node in the network with this particular MIP the sensor node keeps waiting for new sensor nodes with the same MIP.

Several topologies arrangements have been studied in order to know how changes the quality of service parameters when the diameter of the topology increases. The QoS parameters, delay, jitter and packet loss, have been measured for each MIP cluster in three experimental conditions: cluster diameter of one hop, two hops and three hops.

Before the wireless sensors start, they have been configured with static IP address and wireless ad-hoc network configuration, wireless channel and interface speed.

IEEE 802.11g standard has been selected as the wireless technology for the wireless sensor nodes.

The four MIPs are simultaneously working in the same WSN. Two audio MIPs have been selected: AUDIO_64K and AUDIO_192K. First, the AUDIO_64K matches the regular audio communications and Audio IP calls performed through the PCM codification standard and the G.711 codec, the most compatible and widely used at all kind of audio applications and protocols. These deliiveries offer a sound quality similar to the quality of a phone line. The AUDIO_192K MIP matches codecs used at high quality audio communications. With this kind of codecs it is possible to deliver music and human voice with nearly perfect quality.

For video deliveries we have choosen two MIPs: VIDEO_1500K and VIDEO_3500K. The first MIP, VIDEO_1500K, has been choosen because it represents the quality for a video delivery performed in high definition TV (HDTV) with 720p format. In the same way, the VIDEO_3500K is included because it is a typical standard delivery for 1080p format in HDTV.

7.5.1 MIP Comparison

The first test bench was set to find out if there are differences between multimedia deveries belonging to different MIPs when they take place over similiar cluster topologies. In order to study the cluster behaviour in terms of QoS parameters for each MIP, the WSN topology was designed with the aim of building four clusters, one cluster for each MIP. In this experimental design the maximum number of hops for every cluster was established at two hops. In order to be able to compare the obtained results for each cluster, severl environment variables and experimental conditions have been controlled: There are the same number of sensor nodes at each cluster, same average distance between sensor nodes in each cluster, only one multimedia delivery is in progress at a time and the noise level at the 2.4 GHz microwave band is measured and controlled. Fig. 7.31 shows the delay measured for four different MIPs through clusters with identical characteristics, but with different MIP settings. It represents the average delay of the last 20 samples received at any time. In order to estimate the average delay, we applied equation 7.1 on the obtained measurements.

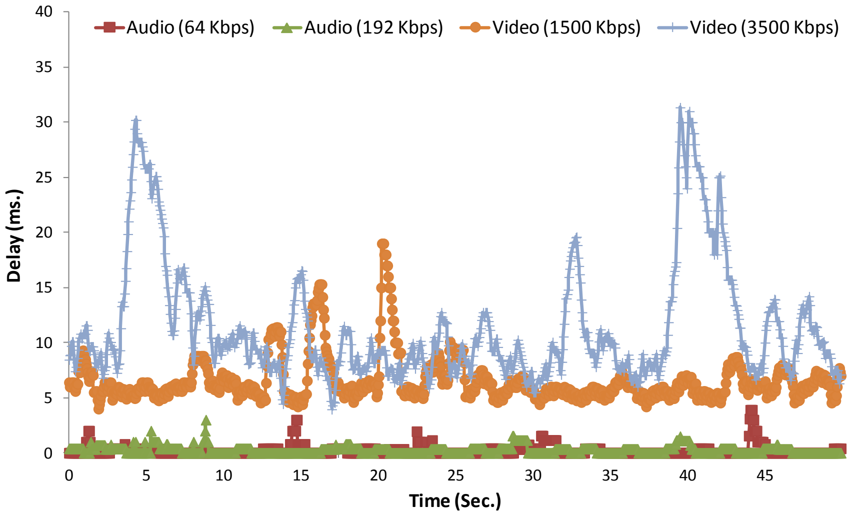

Figure 7.31. Delay as a function of the time for each MIP

Both studied audio codecs have obtained similar delay results. Figure 7.31 shows that the average delay remains below 5 miliseconds when audio is delivered. This results indicate that the quality of audio transmission can be performed over this cluster without any loss of quality, even high quality audio with 192 Kbps. Results for video codecs seem not to be as good as for audio codecs. However, the average delay for video delivery is always below 30 miliseconds (there are two peaks of about 30 milliseconds at the 5^{th} second and at the 40^{th} second) and these values are enough to guarantee an excellent quality on video regular communications.

For the same number of hops, we observe that the delay is rising when the bandwidth spent for multimedia communication through the cluster grows. The behaviour of audio codecs compared to video codecs are clearly differents. In order to determine if there is a significant difference between both audio codecs and between both video codecs, we need to make the statistical analysis of the experimental data. Table 7.5 shows the analysis results. The 99% confidence interval was calculated for an average delay of each experimental condition (α=0,01). In order to establish relationship between each serie, three null hypothesis were assumed: There are not differences between audio and video measures, there are not differences between two audio measures with different bandwidth consumption and there are not differences between two audio measures with different bandwidth consumption.

Table 7.5. Estadistic values and confidence interval for delay as a fucntion of the MIP

MIP	Parameters						
	N	μ (ms)	σ (ms)	Min (ms)	Max (ms)	Confidence interval (ms)	
AUDIO_64K	1000	0.276	0.696	1	4	0.204	0.348
AUDIO_192K	1000	0.205	0.282	1	4	0.175	0.234
VIDEO_1500K	1000	6.588	2.089	4	19	6.370	6.805
VIDEO_3500K	1000	11.461	5.509	5	31	10.887	12.034

As expected, mean delay values are significantly different when any audio codec is compared with any video codec, so we can completely reject the null hypothesis and accept the alternative hypothesis: Difference between delay of audio and video MIPs have statistical significance. In the same way, when the mean delay for both video MIPS are compared a statisticall siginificant difference can be concluded. However, when audio MIPs with different bandwith consumption, 64 Kbps and 192 Kbps, are compared, it is not possible to deduce any significant difference because the mean delay of one audio MIP is inside the confidence interval of the other MIP. In the last case, it is not possible to reject the null hypothesis at least with p=0.01.

Figure 7.32 shows the jitter obtained in the experimental tests. As it happens with the delay results, we can see that the jitter for audio cluster is significantly lower than the jitter for video delivery. The second important result is that all multimedia delivery have jitter values below 15 miliseconds. Only few samples are over the 10 miliseconds. The quality of a multimedia communication can be affected by jitter values when they are as low as 20 or 30 miliseconds, but it is possible to easily manage a jitter value of 15 miliseconds building a buffer in the receiver side to eliminate its harmful effect.

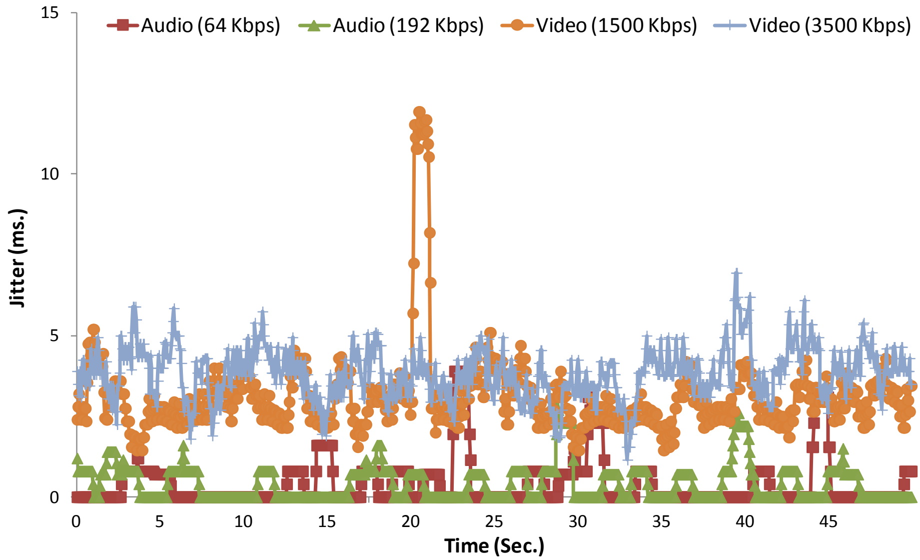

Figure 7.32. Jitter as a function of time for each MIP studied

Data was analyzed to know if there are some significant differences between two similar MIPs, i.e. two audio MIPs or two video MIPs. Table 7.6 shows the statistical parameters for each data serie with α=0.01. Statistical inference has been conducted as the previous delay analysis. Mean jitter values for AUDIO_64K and AUDIO_192K are very similar. Even the AUDIO_192K shows a mean jitter a bit bigger than AUDIO_64K. However, mean value of the first data serie is included at the confidence interval of the second, and viceversa. Null hypothesis cannot be rejected and it is not possible to deduce any difference between both audio data series. By contrast, differences between mean jitter for both video MIP can be accepted. Null hypothesis is rejected in this case. Moreover, the null hypothesis is rejected between the mean jitter values of audio and video MIPs

Table 7.6. Estadistic values and confidence interval for Jitter as a fucntion of the MIP

MIP	Parameters						
	N	μ (ms)	σ (ms)	Min (ms)	Max (ms)	Confidence interval (ms)	
AUDIO_64K	1000	0.412	0.784	0	4	0.348	0.475
AUDIO_192K	1000	0.410	0.556	0	3	0.365	0.455
VIDEO_1500K	1000	3.104	1.315	1	12	2.996	3.211
VIDEO_3500K	1000	3.824	0.786	1	7	3.760	3.888

Based on these results, we can conclude that, in this experimental setup, delay and jitter parameters obtained using different video MIPs are different. Moreover, obtained QoS parameters of video MIPs are different than the QoS parameters of audio MIPs. These results confirm the benefits to divide the whole WSN into several clusters based on MIP configuration. Clusters with multimedia video traffic have a different QoS behaviour as a function of the features of the video delivery and they are also different than the audio delivery. Keeping separate multimedia flows through the MIP architecture allows the network to improve the delay and jitter parameters for multimedia deliverey with low requeriments.

7.5.2 Cluster Comparison

In this second experiment we have studied the number of hops in the cluster. In order to perform this study only one MIP was selected VIDEO_1500K. The WSN topology was modified to achieve three different cluster diameters. Multimedia delivery was always performed through the maximum number of hops allowed in each cluster topology. The number of hops selected for the three experiments were: One, two and three hops.

Figure 7.33 shows the results obtained for the delay as a fuction of the time in the three cases. The main result was that the delay is worse when the number of hops increases, but three hops case has very high peacks, which might be taen into account. Delay for 1 hop delivery is minimal; values were only a few miliseconds above zero, and there was not any big value in the whole serie, all values were below 100 miliseconds. Delay for 2 hops condition moves between 5 and 10 miliseconds; there were some peaks on the graph, however they are small size. Finally, delay for 3 hops transmission was biggest with values between 10 and 20 miliseconds; there are also many peaks with mean values above 70 miliseconds. Multimedia delivery can be optimal with values of up to 150 miliseconds, above this limit quality of service would be decreased. However, it should be noted that the measured delay is only the delay introduced by the sensor nodes transmission on the cluster, but in a real case there are other processes and transmissions out of the cluster that need to be considered to calculate the final delay.

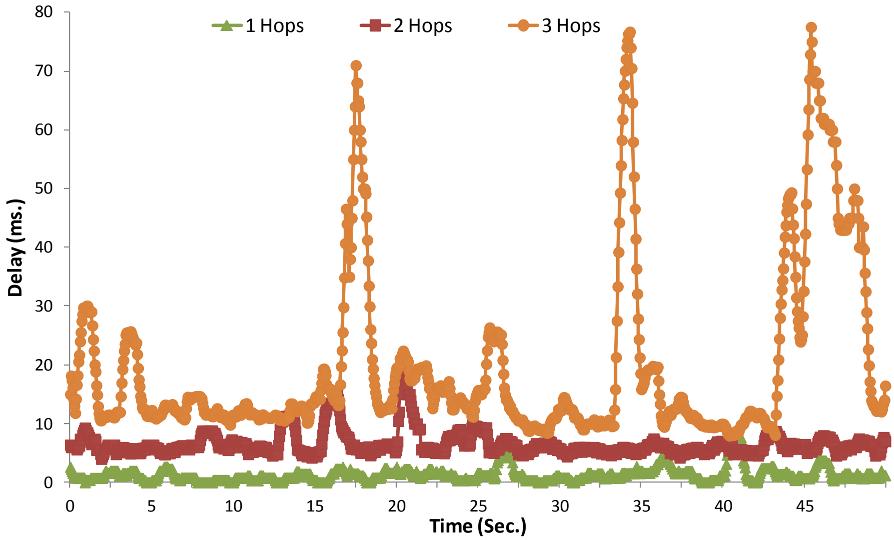

Figure 7.33. Delay as a function of time for different number of hops

In order to corroborate the correct interpretation of these results, a statistical analysis was performed. Table 7.7 shows the statistical analysis. In the inference analysis, α=0.01 is asummed and the 99 % confident interval was calculated for each experimental condition. Two null hypothesis have been stated: There are no differences between the mean delays in the cluster with one hop and the cluster with two hops, and there are no differences between mean delay in the cluster with one hop and the cluster with two hops.

Table 7.7. Estadistic values and confidence interval for delay as a function of the number of hops

HOPS	Parameters						
	N	μ (ms)	σ (ms)	Min (ms)	Max (ms)	Confidence interval (ms)	
1	1000	1.374	1.269	0	8	1.271	1.477
2	1000	6.589	2.089	4	19	6.419	6.759
3	1000	19.797	15.119	8	78	18.565	21.028

Mean delay of each serie is outside the confidence interval of the remaining cases. Both null hypothesis can be fully rejected with 99 % probability. It is possible to affirm that the mean delay value through a two hops cluster is bigger than through a one hop cluster, and the mean delay value through a three hops cluster is bigger than through a two hop cluster.

Jitter values are showed in Figure 7.34. We can see that the values for data series of 1 and 2 hops are very similar, with an average below 5 miliseconds. Otherwises, the 3 hops serie shows higher values, around 10 miliseconds, but it is always below 15 miliseconds. These jitter results have been obtained through the control of some experimental conditions: There was only one delivery, reduced noise level and so on. But in a real environment there are a lot of variables that can affect the multimedia delivery. Thus, a 10 miliseconds level of jitter obtained in these ideal conditions should be interpreted with caution.

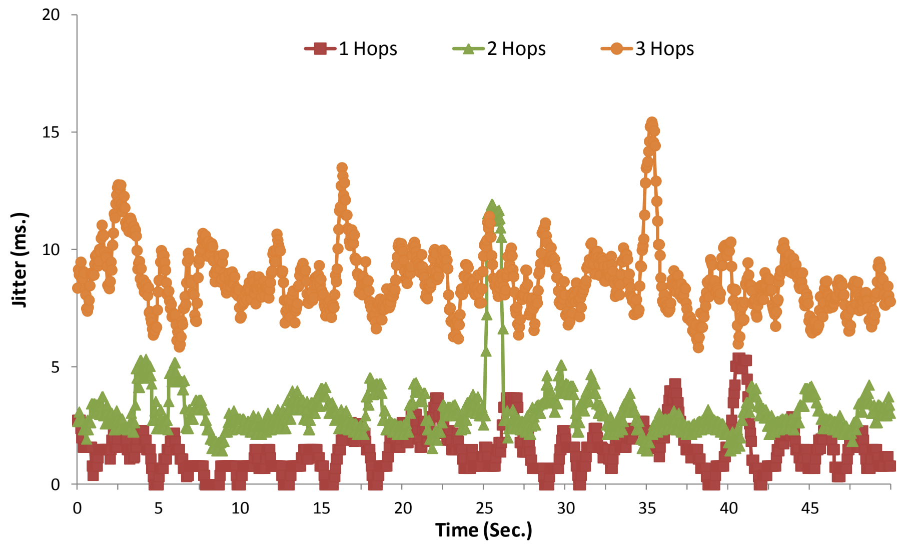

Figure 7.34. Jitter as a function of time for different number of hops

Jitter measures for one and two hops are very similar and we need to make the statistical inference analysis to know the relationship between these data series. The analysis is conducted following the same criteria than the previous delay analysis. It is shown in Table 7.8.

Table 7.8. Estadistic values and confidence interval for jitter as a function of the number of hops

HOPS	**Parameters**						
	N	**μ (ms)**	**σ (ms)**	**Min (ms)**	**Max (ms)**	**Confidence interval (ms)**	
1	1000	1.485	0.974	0	5	1.406	1.564
2	1000	3.118	1.336	1	12	3.009	3.227
3	1000	8.709	1.387	6	15	8.596	8.822

As it happens in the case of delay, null hypothesis can be rejected and the alternative hypothesis is accepted. Mean jitter values for 2 hops serie data is significantly bigger than for 1 hop, and mean jitter value for 3 hops is bigger than for 2 hops.

From these results we can conclude that, in this experimental environment, the cluster diameter may negatively affect the multimedia traffic QoS parameters. The proposal architecture and the developed network protocol can improve QoS parameters by minimizing the maximum number of hops into the cluster.

7.5.3 Packet Loss study

Table 7.9 shows packet loss percentages for each experimental case. We can see that there are very few percentage of lost packets when there is only one hop in the WSN. When the topology becomes more complex, and packets have to make two hops through the WSN, packet loss starts to take relevant values, although that only happens in clusters with assigned video MIP. Video MIP spends an amount of bandwidth between 10 and 50 times thee Audio MIP, so the probability of colision on the wireless network grows. When the number hops rises to three, significally packet loss takes place even for audio MIPS with 64 Kbps and 192 Kbps. We can see that video delivery through three sensor nodes and 3500 Kbps bandwidth, equivalent to HDTV at 1080p, produces over 1% of packet loss. As a function of the codec used for video delivery, this percentage of packet loss can decrease drastically the Quality of Experience (QoE) of the end user.

Table 7.9. Packet loss percentage

Packet Loss	AUDIO_64K	AUDIO_192K	VIDEO_1500K	VIDEO_3500K
1 Hop	0.00%	0.00%	0.00%	0.00%
2 Hops	0.00%	0.00%	0.04%	0.10%
3 Hops	0.19%	0.26%	0.51%	1.30%

The conclusion that we can extract from these results is that loss packet parameter can become a decisive QoS parameter that must be considered when the number of hops is equal o bigger than three hops and the spent multimedia delivery bandwdith is high. MIP based cluster architecture can help by two ways: Limiting the number of hops into a specific cluster and isolating heavy multimedia traffic into a separate cluster in order to improve QoS parameters of the other clustes.

7.6 Conclusion

In this section we have included the performance test and experimental measurements undertaken to validate our proposed system.

On one hand, we have measured QoS for VoIP deliveries, using G.729 and G.711 audio codecs, for 1, 2, 3 or 4 hops. In G.729 there are minimal differences between obtained results. The bandwidth consumed by the communication channel remains constant over time and jitter and latency have negligible values which is good for multimedia purposes. We have obtained similar results for G.711 codec. Despite it consumes three times more bandwidth, all QoS parameters (bandwidth, packet loss, jitter and delay) kept minimum values. Video streaming is more challenging problem than audio streaming. It requires a considerable bandwidth to provide enough QoS. Higher bandwidth is required when there are more resolution and video quality. For low video resolution (600 Kbps) we can observe that jitter, delay and packet loss have acceptable values when multimedia packet makes only one or two hops. But, delay and jitter increase significantly when packet requires to jump three or four times over the wireless ad hoc network to reach the destination. Regardless, the values obtained in the test show that it is possible to deliver video streaming with this low resolution. Latency takes values between 0 and 2 seconds and jitter between 0 and 500 ms. The negative effect of these values can be eliminated buffering the video stream at the destination. For video streaming in the range of 900 to 1800 Kbps, QoS parameters show that it is not possible to deliver video traffic for 3 or 4 hops with acceptable quality level. There are values higher than several seconds for latency and jitter. Moreover, packet loss rises quite high. But, with only one or two hops we can keep video communication with good quality.

On the other hand we have shown how QoS parameters and the multimedia codec characteristics affect the topology of the cluster. We have observed that the cluster diameter affects severely to the delay and jitter.

Morevoer, when providing fault torelance, the measured QoS parameters indicate that both unidirectional and bidirectional multimedia services will not be interrupted at any time even when a node fails. Nevertheless, two relevant restrictions must be considered, mainly in the *FSP*:

1. Jitter should be reduced by configuring a 20 – 60 ms buffer at the end user side

2. Lost packets values for high bandwidth consumption codec can reach an excessive value.

On the other hand, the limited time interval when the *FSP* is used and the low convergence time, justifies the ability of the new fault tolerant mechanism to improve the multimedia communication.

Furthermore, we have developed a new communication protocol that creates ad hoc clusters based on the multimedia flow features that are delivered inside the WSN. We have measured several cases in a test bench with real devices. We have proved that the protocol is able to achieve the adequate values of QoS parameters for different MIPs.

All these experimental results have been included in research papers that have been published in international journals with ISI Thomson Impact Factor. They have been listed in the section “publications derived from the PhD”, in the conclusion chapter.

Chapter 8. Conclusion and Future Research

8.1 Introduction

In this chapter, the conclusion is drawn. It has been a long process full of challenges and achievements since I began my first lectures of the ad hoc networks field. At this time, I have had to learn about experimental procedures, methodological issues, mathematical foundations, network programming, scientific communication and a lot of technical knowledge on wireless networks and protocols.

It has been a learning stage that has greatly contributed to my personal and professional life. From the point of view of my personal life has allowed me to see the world through the eyes of a scientist with critical skills to better understand the environment around me and appreciate details that otherwise would have gone unnoticed. In my role as a teacher, the learned techniques and the deeper knowledge on several matters have become me in a better professional capable of transmit to my students how work the real networks.

Many of the problems I have had to face at this time I cannot overcome them without the invaluable help of my colleagues in the working group I have had the honour of participating. There have always been a lot of ideas emerging that have contributed to continuously modify and improve my work.

8.2 Conclusion

The overall conclusion of this Ph. D. thesis can be summarized as follows:

- We have managed to significantly improve the levels of Quality of Service and Quality of Experience perceived by the end user in the transmission of multimedia traffic over wireless ad hoc networks. An architecture to organize the structure of the network in clusters has been designed, then, a protocol to implement this model has been developed. It is possible to achieve appropriate quality levels for each kind of transmission and, also, it is possible to keep this quality over time, through resource reservation mechanisms.

This dissertation has collected the results of the research conducted to develop a new communication protocol in order to achieve the transmission of multimedia traffic over wireless ad hoc networks, maintaining appropriate quality of service for each individual flow.

In this section, the set of results that have been obtained following the activities carried out in the work plan for each of the research phases are summarized and discussed.

The first phase was dedicated to the collection of preliminary information:

- A study of the state of the art in multimedia transmission over wireless ad hoc networks has been performed following the current scientific literature. The amount of publications addressing this specific issue is relatively small, and mainly they are general proposals to be used over different types of networks, but they are not designed to manage the specific wireless ad hoc networks problems, at most, these networks are only one more of they can be used. However, these proposals do not handle the specific information coming from device and network settings in these networks.

- The behavior of multimedia streams have been studied when they are carried over wireless networks using different topological settings, either ad hoc self-organizing networks or centralized networks that use smart central devices to manage the whole network, as an access point in IEEE 802.11 wireless networks.

- Inside the wireless ad hoc networks, the study focused on multihop networks, where the multimedia packets are sent following routes configured so static or learned through a routing protocol. The routing protocols can use different available algorithms to provide the best path based on criteria as diverse as the power transmission of the device, the geographical location, the battery charge or the shortest path measured by the hop count.

- Measures of quality of service of traffic network were obtained on several communication places: origin, destination and intermediate route hops between them. The measured parameters were: latency, jitter and packet loss. From these collected data in the laboratory, the correlation between QoS parameters and laboratory setting changes were searched.
- It was demonstrated that the internal organization of the nodes within the ad hoc network and the decision algorithm used by the routing protocol are two of the most correlated factors with QoS parameters.
- In general, the published proposals that have been analyzed and the routing protocols studied in the laboratory do not take into account the QoS parameters. Thus, they are not able to distinguish multimedia traffic from other kind of traffic.
- The quality of experience (QoE) was measured independently. The QoE results obtained were similar to QoS expectations.
- Experimental results show that, even flows that do not require excessive bandwidth resources, such as audio calls from IP telephony, suffer an important quality degradation when the when the number of nodes on the path increases.

The second phase was focused on the architecture design to provide the required components to achieve the main objective of this thesis and to support the development of a functional protocol:

- Various proposals for the organization of ad hoc wireless networks with the aim of improving the quality of multimedia content have been ellaborated and published in this stage.
- First models were created to be applied on single set of nodes, which can shape a wired network or a wireless network. In these preliminary models the main point of interest is how the parameters of the nodes, such as their location or their capacity are used to establish the topology.
- From the above results, several new models were presented. They generalize the architecture to be applied to a set of networks, and not a one only network. The algorithms and procedures for the interconnection between them were detailed. One of these models was specificly designed to manage content distribution networks in the Internet service providers so they can improve the service provided to their customers.

- A proposed architecture focusing on wireless ad hoc network was described in detail. Based cluster organization was chosen as the most appopriate mechanism for this type of networks. Thus, the number of hops in the path is always minimized and the optimal route for multimedia traffic is choosen.
- Several node roles are defined into the wireless ad hoc network, although the descentralized essence of ad hoc networks remains. Communication between different clusters and with external content distribution networks is allowed.
- Required processes used to build the topology based on clusters are defined in this point. This Init Multimedia Profile (MIP) definition is introduced as a container with distinctive information. The choice of the MIP used by a network node determines how the clusters will be build. The MIP imposes an important restriction to establish new neighbour connections: only nodes with the same MIP, and therefore with similar characteristics and resources, can share a same cluster. The number and type of multimedia flows managed for an specific node is also limited. When the node runs out their resources it denies new multimedia connections to guarantee the quality of current transmissions. Then, routing algorithm looks for alternativess routes.
- The optimal path calculation is performed independently for each individual multimedia stream, so flows from the same origin and to the same destination may use different routes. Thus, bottlenecks and congestioned links can be avoided. The sources of information used by decision algorithms are the multimedia flow MIP, the nodes MIP on the path and the network and devices status measured in real time.
- Resource reservation mechanisms are incorporate to the original architecture to ensure the resources remain available until the user decides to end the communication.
- A simulation application was programmed to validate the proposal.
- A communications protocol was designed following the proposed model. It implements the architecture functions in order to test both in laboratory environments and real networks. Elements of the protocol header, state tables, types of messages and finite state machine have

been defined according to the indications of the architecture specifications.

- The programmed application of the protocol was run over different scenarios of wireless ad hoc networks mounted in the test laboratory. Measures of performance, reliability, quality of service and quality of experience were taken and analyzed.
- Obtained results show how the proposed protocol organizes the network topology and manages the available resources to achieve the main thesis objective: multimedia transmission over ad hoc wireless networks with quality of service and quality of experience levels similar what the wired network gets.
- Several constraints are detected. The number of concurrent connections supported by a given node in the network is lower than on wired network. Also, maximum number of hops on the route that can be performed on wireless network depends on flow characteristics and if an audio or video codec is used.

Finally, the third phase of the research provides the next results:

- Attention has focused on a specific type of ad hoc wireless networks: wireless sensor networks (WSN). The use of this kind of networks is increasingly widespread.
- Sensors devices offer additional constraints that must be considered: power transmission, battery life, CPU and RAM resources.
- Usually, it is not possible to control key topology factors, such as geographical location, because the network is initially deployed for other different uses than multimedia transmissions.
- Multimedia transmissions over such kind of networks show worse results than in other ad hoc networks. The additional constraints can be tackled and overtaken if specific characteristics are taken into account and are incorporated in the general model.
- The scalability of the architecture developed in the previous phase allows it to adapt to the specific environment of the sensor networks by adding some new message types. Processes and roles are kept identical and decision algorithms are modified to incorporate new parameters from sensor devices.

- As in the previous phase, the research process is completed by programming an application in order to implement the protocol features and test them over real sensor networks.
- The obtained results for QoS parameters of multimedia transmissions when the protocol is running are really similar to those obtained above with the ad hoc network general-purpose protocol.

The general results for the whole research process carried out in this doctoral Thesis, can be summarized in the following three points:

1. A new network architecture that provides the ability to transmit different types of multimedia traffic through ad hoc wireless networks has been proposed and evaluated. It is a scalable model that offers reliability and resources guarantee. Results obtained in the simulations and experimentation have validated the improvement achieved with the proposed model.
2. A protocol implementing the architecture designed has been developed. The protocol takes advantage of all the features of the ad hoc networks for decision-making. Specific kind of networks, such as wireless network sensor, can be easily incorporated in the model. The protocol has been validated with the development of an application that has been run on different types of network settings and devices. Laboratory results show the potential of wireless ad hoc networks, including the case of wireless sensor networks, for transmission of multimedia traffic.
3. As a result of the research, several papers have been published on scientific journals with impact factor and new research lines have been opened.

From the initial specific objectives, the following conclusion can be drawn:

- There is currently no standard solution to send multimedia traffic over ad hoc wireless networks with guaranteed QoS or QoE end-to-end.
- In a wireless network, to receive audio or video streaming, in order to guarantee the necessary resources and to generate an appropriate logical network topology, decision algorithms must weigh both the characteristics of multimedia flows that circulate on the Internet as resources available at each of the nodes and their physical characteristics.

- On wireless ad hoc network, the quality of service should be controlled from the initial formation of the internal structure of the network. I this thesis Ph. D. thesis a cluster based on QoS solution has been developed and successfully tested.
- A modular architecture has been designed to be used as a reference model for protocols and multimedia applications over ad hoc networks. This is a scalable model that will add new elements in the future either to optimize performance in specific networks or to generalize its use to other characteristics of different networks..
- A protocol has been created and an application developed to validate in the laboratory the investigation results. The results confirm that the new proposed architecture can improve QoS and QoE in multimedia connections over ad-hoc wireless networks.
- Finally, the model designed and the protocol and the application developed have also tested on wireless sensor networks, with successful results.

8.3 Future Research

From the results obtained in this Ph. D. thesis, can be extracted different research lines to improve the operation of the proposed architecture. Some of the possibilities are:

- Increase the number of variables included in the MIP container to improve decision algorithms. These factors include node mobility and energy consumption.
- Introduce mechanisms to manage dynamic queues in order to increase the number of concurrent multimedia connections that can be supported by nodes with high congestion level.
- Add security improvements at the protocol by using authentication, encryption and data integrity mechanisms on sent and received messages.

- Make more performance tests over data networks in production, both private and public, mainly when multimedia applications need to compete with other network applications for the available bandwidth.

Moreover, results can be generalized in order to fit:

- Other kind of networks that ad hoc wireless that also show troubles for multimedia traffic transmissions, such as wired network with huge congestion level, long distance point to point links or heterogeneous mobile networks.
- Different applications in addition to multimedia traffic, such as signaling data, industrial traffic or real time traffic from personal devices.

8.4 Publications derived from the PhD

Next papers are derived from the research presented in this dissertation or very related with it. The papers that are directly related with the dissertation are the following papers:

Title:	**A New Multimedia-Oriented Architecture and Protocol for Wireless Ad Hoc Networks**
Authors:	**Juan R. Diaz**, Jaime Lloret, Jose M. Jiménez and Mohammed Hammoumi
Published in:	International Journal of Ad Hoc and Ubiquitous Computing, Volume 16. Issue 1. pages. 14-25 ISSN: 1743-8233 Impact Factor: 0.511
Year:	2014

Title:	**MWAHCA: A Multimedia Wireless Ad Hoc Cluster Architecture**
Authors:	**Juan R. Diaz**, Jaime Lloret, Jose M. Jimenez, Sandra Sendra
Published in:	The Scientific World Journal, Volume 2014, Article ID 913046, 14 pages ISSN: 1537-744X Impact Factor: 1.730
Year:	2014

Title: **A QoS-Based Wireless Multimedia Sensor Cluster Protocol**
Authors: **Juan R. Diaz**, Jaime Lloret, Jose M. Jimenez, Joel J. P. C. Rodrigues
Published in: International Journal of Distributed Sensor Networks
Volume 2014, Article ID 480372, 17 pages
ISSN: 1550-1477
Impact Factor: 0.727
Year: 2014

Title: **Fault Tolerant Mechanism for Multimedia Flows in Wireless ad hoc Networks Based in Fast Switching Paths**
Authors: **Juan R. Diaz,** Jaime Lloret, Jose M. Jimenez, Sandra Sendra, Joel J. P. C. Rodrigues
Published in: Mathematical Problems in Engineering
ISSN: 1024-123X
Impact Factor: 1.383
Year: 2014

During last years I have collaborated in the development of several models, proposals and experimental set-ups which have served as first approach for the final purpose of this dissertation. They have helped me to build the foundations on which the final architecture is grounded. These preliminary works have not only contributed substantially to the progress of this research, but also opened new lines of related research that have subsequently been or will be developed by other members of the research group with whom I have been collaborating. The published papers corresponding to this work are presented in a prominent place within this thesis.

Title: **An Architecture to Connect Disjoint Multimedia Networks Based on node's Capacity**
Authors: Jaime Lloret, **Juan R. Diaz**, Jose M. Jimenez, Fernando Boronat
Published in: Lecture Notes in Computer Science.
Volume 4261. Pags. 890-899
ISSN: 0302-9743
Year: 2006

Title: **A Group-based Content Delivery Network**
Authors: Jaime Lloret, Miguel Garcia, Diana Bri, **Juan R. Diaz**
Published in: ACM/IEEE International Symposium on High Performance Distributed Computing 2008 (HPDC 2008)
ISBN: 978-1-59593-997-5
doi>10.1145/1384209.1384217
Year: 2008

Title: **Study and performance of a group-based content delivery network**
Authors: Jaime Lloret, Miguel Garcia, Diana Bri, **Juan R. Diaz**
Published in: Journal of Network and Computer Applications,
Volume 32, Issue 5, pages 991-999
ISSN: 1084-8045
Impact Factor: 1.111
Year: 2009

Title: **A Cluster-Based Architecture to Structure the Topology of Parallel Wireless Sensor Networks**
Authors: Jaime Lloret, Miguel Garcia, Diana Bri and **Juan R. Diaz**
Published in: Sensors 2009,
Volume 9, Issue 12, pages. 10513-10544
ISSN: 1424-8220
Impact Factor: 1.821
Year: 2009

Ther are other papers published which have contributed less to the base of this Ph.D., but they should be mentioned.

Next paper helped me to understand the design of wireless networks in harsh environments.

Jaime Lloret, Pedro V. Mauri, Jose M. Jimenez and Juan R. Diaz, 802.11g WLANs Design for Rural Environments Video-surveillance, International Conference on Digital Telecommunications (ICDT'06), Cap Esterel (France). August 26-31, 2006

The following paper was useful for the understanding of communications in multicast groups for multimedia delivery.

J. R. Diaz, J. Lloret, J. M. Jiménez, Reliability In Multicast Communications, WSEAS Transactions on Computers, Issue 6, Volume 3, Pp. 2031-2036. December 2004

The following paper was the basis of the design and development of the designed fault toletant algorithm in this thesis.

Jaime Lloret, Juan R. Diaz, Fernando Boronat and Jose M. Jiménez, A Fault-Tolerant P2P-based Protocol for Logical Networks Interconnection, International Conference on Networking and Services (ICNS'06), Silicon Valley (USA), July 16-22, 2006.

In this paper I applied the knowledge about audio streaming adquired during the realization of the thesys.

Rubén Tortosa, Jose M. Jiménez, Juan R. Diaz, Jaime Lloret, Optimal Codec Selection Algorithm for Audio Streaming, IEEE Global Communications Conference Workshops (IEEE Globecom 2014), Austin, Texas (USA), December 8-12, 2014.

In this paper I applied the knowledge designing ad hoc protocols adquired during the research of my PhD

Carlos Cambra Baseca, Juan R. Diaz, and Jaime Lloret, Communication Ad Hoc Protocol for Intelligent Video Sensing using AR Drones, IEEE Ninth International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2013), Dalian (China), December 11-13, 2013

References

[1] Miguel Garcia, Diana Bri, Sandra Sendra and Jaime Lloret, Practical Deployments of Wireless Sensor Networks: a Survey, Journal On Advances in Networks and Services, Vol. 3, Issue 1&2, pp. 170-185, July 2010.

[2] Diana Bri, Miguel Garcia, Jaime Lloret and Petre Dini, Real Deployments of Wireless Sensor Networks, The Third International Conference on Sensor Technologies and Applications (Sensorcomm 2009), pp. 415-423, Atenas (Grecia), June 18-23, 2009.

[3] Sandra Sendra, Jaime Lloret, Miguel Garcia and Jose F. Toledo, Power saving and energy optimization techniques for Wireless Sensor Networks, Journal of Communications, Vol. 6, Issue 6, pp. 439-459, August 2011.

[4] F. Akyildiz, W. Su, Y. S. Subramaniam and E. Cayirci, A Survey on Sensor Networks, IEEECommunications Magazine, Vol. 40, Issue 8, pp. 102-114, August 2002.

[5] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin and Aamir Hassan, Vehicular ad hoc networks (VANETS): status, results, and challenges, Telecommunication Systems, Vol. 50, Issue 4, pp 217-241, August 2012.

[6] Mohammad Jalil Piran, G. Rama Murthy and G. Praveen Babu, Vehicular Ad Hoc and Sensor Networks; Principles and Challenges, International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC), Vol. 2, Issue 2, pp. 38-49, June 2011.

[7] Jaime Lloret, Miguel García, Fernando Boronat and Jesus Tomás, MANET protocols performance in group-based Networks, 10th IFIP International Conference on Mobile and Wireless Communications Networks (MWCN 2008), Toulouse (Francia), Vol. 284, pp. 161-172, September 30 - October 2, 2008.

[8] Raquel Lacuesta, Jaime Lloret, Miguel Garcia and Lourdes Peñalver, Two Secure and Energy-Saving Spontaneous Ad-Hoc Protocol for Wireless Mesh Client networks, Journal of Network and Computer Applications, Vol. 34, Issue 2, pp. 492-505, March 2011.

[9] Jimmy Jernberg, Vladimir Vlassov, Ali Ghodsi, Seif Haridi, DOH: A Content Delivery Peer-to-Peer Network, European Conference on Parallel Computing, (EUROPAR 2006), pp 1026-1039, Dresden (Germany), August 29 – September 1, 2006.

[10] Alan Kin Wah Yim and Rajkumar Buyya, Decentralized media streaming infrastructure (DeMSI): An adaptive and high-performance peer-to-peer content delivery network, Journal of Systems Architecture: the EUROMICRO Journal, Vol. 52, Issue 12, pp 737-772, December 2006.

[11] Antonella Di Stefano and Corrado Santoro, An economic model for resource management in a Grid-based content distribution network, Future Generation Computer Systems, Vol. 24, Issue 3, pp 202-212, March 2008.

[12] Giancarlo Fortino and Wilma Russo, Using P2P, GRID and Agent technologies for the development of content distribution networks, Future Generation Computer Systems, Volume 24, Issue 3, pp 180-190, March 2008.

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

[14] David Andersen, Hari Balakrishnan, Frans Kaashoek and Robert Morris, Resilient Overlay Networks, 18th ACM Symposium on Operating Systems Principles (SOSP 2001), Banff (Canada), October 2001.

[15] Sylvia Ratnasamy, Mark Handley, Richard Karp and Scott Shenker, Topologically-Aware Overlay Construction and Server Selection, IEEE INFOCOM 2002, Vol. 3, pp 1190-1199, New York (USA), June 23-27, 2002.

[16] Jian Ni Tsang, Danny H. K. Tsang, Ivan S. H. Yeung and Xiaojun Hei, Hierarchical content routing in large-scale multimedia content delivery network, IEEE International Conference on Communications (ICC '03), Vol. 2, pp. 854- 859, Anchorage (USA), May 11-15, 2003.

[17] Jian Ni, Danny H. K. Tsang, Large-scale cooperative caching and application-level multicast in multimedia content delivery networks, IEEE Communications Magazine, Vol. 43, Issue 5, pp. 98- 105, May 2005.

[18] Jane Y. Yu and Peter H. J. Chong, A survey of clustering schemes for mobile ad hoc networks, IEEE Communications Surveys & Tutorials, Vol. 7, Issue 1, pp. 32- 48, First Qtr. 2005.

[19] Samba Sesay, Zongkai Yang and Jianhua He. A Survey on Mobile Ad Hoc Wireless Network. Information Technology Jorunal, Vol. 3, Issue 2, pp. 168-175, 2004

[20] Kui Wu and Janelle Harms, QoS Support in Mobile Ad Hoc Networks, Crossing Boundaries an Interdisciplinary Journal, pp. 92-107, March 2001.

[21] H. Xiao, W.K.G. Seach, A. Lo and K.C. Chua, A Flexible Quality of Service Model for Mobile Ad Hoc Networks, Vehicular Technology Conference (VTC 2000), Vol. 1, pp. 445-449, Tokyo (Japan), May 15-18, 2000.

[22] L. Hanzo and R. Tafazolli, A survey of QoS routing solutions for mobile ad hoc networks, IEEE Communications Surveys & Tutorials, Vol. 9, Issue 2, pp. 50-70, Second Quarter 2007.

[23] Mohammed Tarique, Anwar Hossain, Rumana Islam and C. Akram Hossain, Issues of Long-Hop and Short-Hop Routing in Mobile Ad Hoc Networks: A Comprehensive Study, Network protocols and Algorithms, Vol. 2, Issue 2, pp. 107-131, June 2010.

[24] Raquel Lacuesta, Miguel Garcia, Jaime Lloret, and Guillermo Palacios, Study and Performance of Ad Hoc Routing Protocols, chapter 4 of the book Mobile Ad hoc Networks: Current Status and Future Trends, CRC Press, ISBN 9781439856505, pp. 71-101, November 2011.

[25] B. Zhang, and H. T. Mouftah, QoS routing for wireless ad hoc networks: problems, algorithms, and protocols, IEEE Communications Magazine, Vol. 43, Issue 10, pp. 110-117. October 2005.

[26] H. Luo, R. Ramjee, P. Sinha, L. E. Li, and S. Lu, UCAN: a unified cellular and ad-hoc network architecture, 9th annual international conference on Mobile computing and networking (MobiCom'03), pp. 353-367, San Diego, California (USA), September 14–19, 2003.

[27] B. Li, and K. H. Wang, NonStop: Continuous multimedia streaming in wireless ad hoc networks with node mobility, IEEE Journal on Selected Areas in Communications, Vol. 21, Issue 10, pp. 1627-1641, December 2003.

[28] P. Sinha, QoS issues in ad-hoc networks. Chapter 8 of the book in Ad Hoc Networks, Springer US, ISBN 9780387226897, pp. 229-247, 2005.

[29] S. Chakrabarti and A. Mishra, QoS issues in ad hoc wireless networks, IEEE Communications Magazine, Vol. 39, Issue 2, pp. 142-148, Febrary 2001.

[30] P. Mohapatra, J. Li, and C. Gui, QoS in mobile ad hoc networks: A survey, IEEE Wireless Communications, Vol. 10, Issue 3, pp. 44-53, June 2003.

[31] A. M. Abbas and O. Kure, Quality of Service in mobile ad hoc networks: a survey, International Journal of Ad Hoc and Ubiquitous Computing, Vol. 6, Issue 2, pp. 75-98, July 2010.

[32] P. H. Chuang, H-K. Wu, and M.-K. Liao, Dynamic QoS allocation for multimedia ad hoc wireless networks. Mobile Networks and Applications, Vol.6, Issue 4, pp. 377-384, August 2001.

[33] R. Zhang, L. Cai, J. Pan and X. S. Shen, Resource management for video streaming in ad hoc networks, Ad Hoc Networks, Vol. 9, Issue 4, pp. 623-634, June 2011.

[34] T. Mehta and Z. Narmawala, Performance Enhancement Of Multimedia Traffic Over Wireless Ad Hoc Networks Using Network Coding, Nirma University

International Conference on Engineering (NUiCONE-2012), pp. 1-6, Ahmedabad (India), December 6-8, 2012.

[35] R. AlTurki and R. Mehmood, Multimedia Ad hoc Networks: Performance Analysis. Computer Modeling and Simulation (EMS ’08), pp 561-566, Liverpool (UK), September 8-10, 2008.

[36] A. Jamali, N. Naja and D. El ouadghiri, Comparative analysis of ad hoc networks routing protocols for multimedia streaming, Computer Modeling and Simulation (EMS 2009), Ouarzazate (Morroco), pp. 381 – 385, April 2-4, 2009.

[37] A. Abdrabou and W. Zhuang, Statistical QoS routing for IEEE 802.11 multihop ad hoc networks, IEEE Transactions on Wireless Communications, 2009, Vol. 8, Issue 3, pp. 1542-1552, March 2009

[38] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and, S. Kotsopoulos, Energy efficient and perceived QoS aware video routing over Wireless Multimedia Sensor Networks, Ad Hoc Networks, Vol. 9, Issue 4, pp.591-607, June 2011.

[39] N. Taing, S. Thipchaksurat, R. Varakulsiripunth and H. Ishii, Routing Scheme for Multimedia Services in Mobile Ad Hoc Network, Information, Communications and Signal Processing, 2005, pp. 11 -15, Bangkok (Thailand), December 6-9, 2005.

[40] A. S. Tai, K. S. Tso and W.H. Sanders, Cluster-based failure detection service for large-scale ad hoc wireless network applications, International Conference on Dependable Systems and Networks ,2004, pp. 805-814, Florence (Italy), June 28- July 1, 2004.

[41] F. Kuhn, T. Moscibroda and R. Wattenhofer, Fault-tolerant clustering in ad hoc and sensor networks, 26th IEEE International Conference on Distributed Computing Systems (ICDCS 2006), pp. 68-68, Lisboa, (Portugal), July 4-7, 2006.

[42] Y. Xue and K. Nahrstedt, Providing fault-tolerant ad hoc routing service in adversarial environments, Wireless Personal Communications, Vol. 29, Issue 3, pp 367-388, May 2004.

[43] H.L. Chao and, C.L. Chang, A fault-tolerant routing protocol in wireless sensor networks, International Journal of Sensor Networks, Vol. 3, Issue 1, pp 66-73, December 2007.

[44] E. Pagani and G.P. Rossi, Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks, Mobile Networks and Applications, Vol.4, Issue 3, pp. 175-192, September 1999.

[45] G. Gupta and M. Younis, Fault-tolerant clustering of wireless sensor networks, IEEE Wireless Communications and Networking (WCNC 2003), Vol. 3, pp. 1579-1584, New Orleans, Lousiana, (USA), March 16-20, 2013.

[46] A. Boukerche, R. Werner Nelem Pazzi and R. Borges Araujo, Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments, Journal of Parallel and Distributed Computing, Vol. 66, Issue 4, pp. 586-599, April 2006.

[47] T. Bheemarjuna Reddy, S. Sriram, B.S. Manoj and C. Siva Ram Murthy, MuSeQoR: Multi-path failure-tolerant security-aware QoS routing in Ad hoc wireless networks, Computer Networks, Vol. 50, Issue 9, pp. 1349-1383, June 2006.

[48] R. Melamed, I. Keidar and Y. Barel, Octopus: A fault-tolerant and efficient ad-hoc routing protocol, Wireless Networks, Vol. 14, Issue 6, pp. 777-793, December 2008.

[49] T. Antoniou, I. Chatzigiannakis, G. Mylonas, S. Nikoletseas and A. Boukerche, A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range, 37th annual symposium on Simulation, pp. 43-52, Washington, DC (USA), April 18-22, 2004.

[50] C. R. Lin and M. Gerla, Adaptive clustering for mobile wireless networks. IEEE Journal on Selected Areas in Communications, Vol. 15, issue 7, pp. 1265-1275. September 2006.

[51] J. H. Ryu, S. Song and D.-H. Cho, New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks, International Conference on Communications (IEEE ICC'01), Vol. 3, pp. 862–866, Helsinki (Finland), June 11-14, 2001.

[52] D. Wei and H. A. Chan, Clustering Ad Hoc Networks: Schemes and Classifications, Third Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06), pp. 920-926, Reston, VA (USA), September 25-28, 2006.

[53] A. A. Abbasi and M. Younis, A survey on clustering algorithms for wireless sensor networks, Computer Communications, Vol. 30, Issues 14-15, pp. 2826-2841, October 2007.

[54] O. Boyinbode, H. Le and M. Takizawa, A survey on clustering algorithms for wireless sensor networks, International Journal of Space-Based and Situated Computing, Vol. 1, Issues 2-3, pp. 130-136, May 2011.

[55] R. Agarwal and M. Motwani, Survey of clustering algorithms for MANET, International Journal on Computer Science and Engineering (IJCSE), Vol. 1, Issue 2, pp. 98-104, 2009.

[56] L. Ramachandran, M. Kapoor, A. Sarkar and A. Aggarwal, Clustering Algorithms for Wireless Ad Hoc Networks, 4th international workshop on

Discrete algorithms and methods for mobile computing and communications (DIALM '00), pp. 54-63, Boston, MA (USA), August 11, 2000.

[57] M. Chatterjee, S. K. Das and D. Turgut, WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks, Cluster Computing, Vol. 5, Issue 2, pp. 193-204, April 2002.

[58] M. Kavitha and K. Karthikeyan, Comparison of data centric protocols for WSN and energy enhanced M-Spin (EEM-SPIN), International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 9, November 2012.

[59] Y. Huang, M. Hsieh and M. Wang, Reliable transmission of multimedia streaming using a connection prediction scheme in cluster-based ad hoc networks, Computer Communications, Vol. 30, Issue 2, pp. 207-496, January 2007.

[60] S. Tang and W. Li, QoS supporting and optimal energy allocation for a cluster based wireless sensor network, Computer Communications, Vol. 29, Issues 13–14, pp. 2569–2577, August 2006.

[61] Denis Rosário, Rodrigo Costa, Helder Paraense, Kássio Machado, Eduardo Cerqueira, Torsten Braun and Zhongliang Zhao, A Hierarchical Multi-hop Multimedia Routing Protocol for Wireless Multimedia Sensor Networks, Network Protocols and Algorithms, Vol. 4, Issue 4, pp. 44-64, December 2012.

[62] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, A Survey and Comparison of Peer to-Peer Overlay Network Schemes, IEEE Communications Surveys and Tutorial, Vol. 7, Issue 2, pp. 72-93, Second Quarter 2005.

[63] A. Vakali and G. Pallis, Content delivery networks: status and trends, IEEE Internet Computing, Vol. 7, Issue 6, pp. 68-74, December 2003.

[64] Shareaza, available at, http://www.shareaza.com/ [Last access: December 11, 2015].

[65] MLDonkey, available at, http://mldonkey.berlios.de/ [Last access: December 11, 2015].

[66] Morpheus, available at, http://www.morpheus.com [Last access: December 11, 2015].

[67] The giFT Project, available at, http://gift.sourceforge.net/ [Last access: December 11, 2015].

[68] Benno J. Overeinder, Etienne Posthumus and Frances M.T. Brazier, Integrating Peer-to-Peer Networking and Computing in the AgentScape Framework, Second International Conference on Peer-to-Peer Computing (P2P'02), pp. 96-103, Linköping (Sweden), September 5-7, 2002.

[69] J. Lloret, F. Boronat, C. Palau and M. Esteve, 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 (ICAS-ICNS 2005), pp. 39, Papeete, Tahiti (French Polynesia), October 23-28, 2005.

[70] Edsger W. Dijkstra and C. S. Scholten, Termination detection for diffusing computations, Information Processing Letters, Vol. 11, Issue 1, pp. 1-4, August 1980.

[71] Yogen K. Dalal and Robert M. Metcalfe, Reverse path forwarding of broadcast packets, Communications of the ACM, Vol. 21, Issue 12, pp. 1040-1048. December 1978.

[72] C. Cramer, K. Kutzner and T. Fuhrmann, Bootstrapping Locality-Aware P2P Networks, 12th IEEE International Conference on Networks (ICON 2004), Vol. 1, pp. 357-361, Singapore (Singapore), November 16-19, 2004.

[73] Beverly Yang and Hector Garcia-Molina, Improving Search in Peer-to-Peer Networks, 22nd International Conference on Distributed Computing Systems (ICDCS'02), pp. 5-14, Vienna (Austria), July 2-5, 2002.

[74] John Apostolopoulos, Tina Wong, Wai-tian Tan and Susie Wee, On multiple decription streaming with content delivery networks, IEEE Infocom 2002, New York, NY (USA), June 23-27, 2002.

[75] Akamai, available at, http://www.akamai.com [Last access: December 11, 2015].

[76] H. Hlavacs, M. Haddad, C. Lafouge, D. Kaplan and J. Ribeiro, The CODIS Content Delivery Network, Computer Networks, Vol. 48, Issue 1, pp. 75-89, May 2005.

[77] A. Dutta and H. Schulzrinne, MarconiNet: overlay mobile content distribution network. IEEE Communications Magazine, Vol. 42, Issue 2, pp. 64-75, February 2004.

[78] D. Verma, Content Distribution Networks. An Engineering Approach, Ed. Wiley, Print ISBN: 9780471443414, New York, NY (USA), February 2002.

[79] G. Pierre and M. van Steen, Design and implementation of a user-centered content delivery network, Third IEEE Workshop on Internet Applications (WIAPP 2003), pp. 42-49, San Jose, CA (USA), June 23-24, 2003.

[80] Michael J. Freedman, Eric Freudenthal and David Mazières. Democratizing Content Publication with Coral, 1st conference on Symposium on Networked Systems Design and Implementation (NSDI '04), pp. 18-18, San Francisco, CA (USA), March 29-31, 2004.

[81] V.S. Pai, L. Wang, K.S. Park, R. Pang and L. Peterson, The dark side of the Web: an open proxy's view, ACM SIGCOMM Computer Communication Review, Vol. 14, Issue 1, pp. 57-62, January 2004.

[82] M. Day, B. Cain, G. Tomlinson and P. Rzewski, RFC 3466: A Model for Content Internetworking (CDI), available at, https://tools.ietf.org/html/rfc3466, 2003. [Last access: December 11, 2015].

[83] A. Wierzbicki, R.Strzelecki, D. Swierczewski and M. Znojek, Rhubarb: a Tool for Developing Scalable and Secure Peer-to-Peer Applications, Second International Conference on Peer-to-Peer Computing (P2P'02), pp. 144-151, Linöping (Sweden), September 5-7, 2002.

[84] Hongjun Liu, Ping Luo and Zhifeng Zeng, Structured hierarchical P2P model based on a rigorous binary tree code algorithm, Future Generation Computer Systems, Vol. 23, Issue 2, pp. 201-208, February 2007.

[85] B. Thallner and H. Moser, Topology control for fault-tolerant communication in highly dynamic wireless networks, Third International Workshop on Intelligent Solutions in Embedded Systems (WISES 2005), pp. 89-100, Hamburg (Germany), May 20, 2005.

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

[87] M. Jiang, J. Li and Y. C. Tay, Cluster Based Routing Protocol (CBRP), available at http://tools.ietf.org/html/draft-ietf-manet-cbrp-spec-01.txt, August 1998. [Last access: December 11, 2015].

[88] R. K. Ghosh, V. Garg, M. S. Meitei, S. Raman, A. Kumar and N. Tewari, Dense cluster gateway based routing protocol for multi-hop mobile ad hoc networks. Ad Hoc Networks, Vol. 4, issue 2, pp. 168-185, March 2006.

[89] T. Ohta, S. Inoue and Y. Kakuda, An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks, Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03), pp. 293-300, Pisa (Italy), April 9-11, 2003.

[90] Y. Jin, L. Wang, Y. Kim and X. Yang. EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks. Computer Networks, Vol. 52, issue 3, pp. 542-562, February 2008.

[91] B. Das and V. Bharghavan, Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets, IEEE International Conference on Communications (ICC'97), Vol.1 1, pp. 376–380, Montreal (Canada), June 8-12, 1997.

[92] P. Basu, N. Khan and T. D. C. Little, A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks, International Conference on Distributed Computing Systems Workshop (IEEE ICDCSW’ 01), pp. 413-418, Phoenix, AZ (USA), April 16-19, 2001.

[93] Qingchun Ren and Qilian Liang, Energy-Efficient Medium Access Control Protocols for Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking, Vol. 2006, Issue 2, pp. 5-5, April 2006.

[94] A. D. Amis and R. Prakash, Load-Balancing Clusters in Wireless Ad Hoc Networks, Third IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET’00), pp. 25-32, Richardson, TX (USA), March 2000.

[95] X. Du and F. Lin, Maintaining Differentiated Coverage in Heterogeneous Sensor Networks. EURASIP Journal on Wireless Communications and Networking, Vol. 2005, Issue 4, pp. 565–572, September 2005.

[96] Y. Zou and K. Chakrabarty, Sensor deployment and target localization in distributed sensor networks. ACM Transactions on Embedded Computing Systems (TECS), Vol. 3, Issue 1, pp. 61-91, February 2004.

[97] Yong Yuan, Min Chen and Taekyoung Kwon. A Novel Cluster-Based Cooperative MIMO Scheme for Multi-Hop Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking, Vol. 2006, Issue 2, pp. 1–9, April 2006.

[98] J. Ibriq, and I. Mahgoud, Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges. Symposium on Performance Evaluation of Computer Telecommucations Systems (SPECTS’04), pp. 759–766, San Jose, CA (USA), July 25-29, 2004.

[99] Keun-Ho Lee, Sang-Bum Han, Heyi-Sook Suh, SangKeun Lee and Chong-Sun Hwang, Authentication Based on Multilayer Clustering in Ad Hoc Networks, EURASIP Journal on Wireless Communications and Networking, Vol. 2005, Issue 5, pp. 731-742, October 2005.

[100] K. Kredo and P. Mohapatra, Medium access control in wireless sensor networks. Computer Networks, Vol. 51, Issue 4, pp. 961-994, March 2007..

[101] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol (OLSR), RFC 3626, available at http://www.ietf.org/rfc/rfc3626.txt, October 2003. [Last access: December 11, 2015].

[102] C. Perkins, E. Belding-royer and S. Das, Ad Hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, available at http://www.ietf.org/rfc/rfc3561.txt, [Last access: December 11, 2015], July 2003.

[103] D. Johnson, Y. Hu and D. Maltz, The Dynamic Source Routing Protocol (DSR) for Mobile Ad-hoc Networks for IPv4, RFC 4728, available at http://www.ietf.org/rfc/rfc4728.txt, February 2007. [Last access: December 11, 2015].

[104] V. Park and S. Corson, Temporally-Ordered Routing Algorithm (TORA) Version 1, Functional Specification, Internet Draft, available at http://tools.ietf.org/id/draft-ietf-manet-tora-spec-04.txt, June 2001. [Last access: December 11, 2015].

[105] Yao-Chung Chang, Zhi-Sheng Lin and Jiann-Liang Chen, Cluster based self-organization management protocols for wireless sensor networks, IEEE Transactions on Consumer Electronics, Vol. 52, Issue 1, pp. 75 – 80, February 2006.

[106] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, 33rd Annual Hawaii International Conference on System Sciences, Vol. 8, Page 8020, Maui, Hawaii (USA), January 4-7, 2000.

[107] D. Bri, H. Coll, M. Garcia and J. Lloret, A Multisensor Proposal for Wireless Sensor Networks, Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008), pp. 270-275, Cap Esterel, Côte d'Azur (France), August 25-31, 2008.

[108] Wireshark Network Protocol Analyzer website, available at http://www.wireshark.org [Last access: December 11, 2015]

[109] P. Krishna P, N. H. Vaidya, M. Chatterjee and D. K. Pradhan, A cluster-based approach for routing in dynamic networks, ACM SIGCOMM Computer Communication Review, Vol. 27, Issue 2, pp. 49-64, April 1997.

[110] M. Bechler, H.-J. Hof, D. Kraft, F. Pählke and L. Wolf, A Cluster-Based Security Architecture for Ad Hoc Networks. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), Vol. 4, pp. 2393-2403, Hong Kong, China, Mar 7-11, 2004.

[111] S. Leng, L. Zhang, H. Fu and J. Yang, A Novel Location-Service Protocol Based on k -Hop Clustering for Mobile Ad Hoc Networks. IEEE Transactions on Vehicular Technology, Vol. 56, Issue 2, pp. 810-817, March 2007.

[112] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin and A. O. Fapojuwo, A centralized energy-efficient routing protocol for wireless sensor networks. IEEE Communication Magazine. Vol. 43, Issue 3, pp. 8-13, March 2005.

[113] T. F. Shih and H. C. Yen, Core Location-Aided Cluster-Based Routing Protocol for Mobile Ad Hoc Networks, 10th WSEAS international conference

on Communications (WSEAS ICC 2006), pp. 223-228, Vouliagmeni, Athens (Greece), July 10-12, 2006.

[114] Y. Wang, H. Chen, X. Yang and D. Zhang, Cluster Based Location-Aware Routing Protocol for Large Scale Heterogeneous MANET, Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007), pp. 363-373, Iowa City, IA (USA), 13-15 August 2007.

[115] C.-Y. Wen and W. A. Sethares, Automatic Decentralized Clustering for Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking. Vol. 2005, Issue 5, pp. 686–697, October 2005.

[116] C. C. Shen, C. Srisathapornphat, R. Liu, Z. Huang, C. Jaikaeo and E. L. Lloyd, CLTC: A Cluster-Based Topology Control Framework for Ad Hoc Networks. IEEE Transactions on Mobile Computing, Vol. 3, Issue 1, pp. 18- 32, January-February 2004.

[117] J. Chen and F. Yu, A Uniformly Distributed Adaptive Clustering Hierarchy Routing Protocol, IEEE International Conference on Integration Technology (ICIT 2007), pp. 628-632, Shenzhen (China), March 20-24, 2007.

[118] O. Younis and S. Fahmy, Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach, Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), Vol. 1, Hong Kong (China), March 7-11, 2004.

[119] T. D. Hollerung, The Cluster-Based Routing Protocol. University of Paderborn. 2003. Winter semester 2003/2004.

[120] Dapeng Wu, Y. Thomas Hou, Wenwu Zhu, Zhihai He, Ya-Qin Zhang, Adaptive QoS control for real-time video communication over wireless channels, International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Vol. 1, Issue 1-2, pp. 27-37, 2005.

[121] CISCO, White paper: Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017, available at Cisco web site: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html. [Last access: December 11, 2015].

[122] CISCO, White paper: Cisco Visual Networking Index: Forecast and Methodology, 2011–2016, available at Cisco web site: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html [Last access: December 11, 2015].

[123] Wireless Intelligence, company reports (2012), Analysis: The top 20 global operator groups by mobile connections (Q2 2012), available at Wireless

Intelligence web site: https://wirelessintelligence.com/analysis/2012/10/the-top-20-global-operator-groups-by-mobile-connections-q2-2012/353/, October 2012. [Last access: December 11, 2015]

[124] Kansas State University, Online press article (September 27, 2012). Doctoral student developing next generation of lithium-ion batteries for longer lasting mobile devices, electric cars, available at ScienceDaily web site: http://www.sciencedaily.com/releases/2012/09/120927152139.htm, [Last access: December 11, 2015]

[125] Northwestern University, Online press article (November 14, 2011), New technology improves both energy capacity and charge rate in rechargeable batteries, available at Northwestern University web site: http://www.northwestern.edu/newscenter/stories/2011/11/batteries-energy-kung.html. [Last access: December 11, 2015]

[126] J. Mueller, T. Magedanz and J. Fiedler, NNodeTree: A Scalable Peer-to-Peer Live Streaming Overlay Architecture for Next-Generation-Networks, Network Protocols and Algorithms, Vol. 1, Issue 2, pp. 61-84, 2009.

[127] C. J. Barenco Abbas, L. J. García Villalba and A. L. Sandoval Orozco, A distributed QoS mechanism for ad hoc network, International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Vol. 11, Issue 1, pp. 25-3, September 2012.

[128] T. B. Reddy, I. Karthigeyan, B. S. Manoj and C. Murthy, Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. Ad Hoc Networks, Vol.4, Issue 1, pp.83-124, January 2006.

[129] Y. Chen, T. Farley and N. Ye, QoS requirements of network applications on the Internet, Information Knowledge Systems Management, Vol.4, Issue 1, pp. 55-76, January 2004.

[130] Raquel Lacuesta, Jaime Lloret, Miguel Garcia and Lourdes Peñalver, A Spontaneous Ad Hoc Network to Share WWW Access, EURASIP Journal on Wireless Communications and Networking, Vol. 2010, Issue May, pp.1-16, April 2010.

[131] Raquel Lacuesta, Jaime Lloret, Miguel Garcia and Lourdes Peñalver, A Secure Protocol for Spontaneous Wireless Ad Hoc Networks Creation, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, Issue 4, pp. 629-641, June 2013.

[132] K. Sahadevaiah and P.V.G.D.P. Reddy, Impact of Security Attacks on a New Security Protocol for Mobile Ad Hoc Networks, Network Protocols and Algorithms, Vol. 3, Issue 4, pp. 122-140, December 2011.

[133] Jaime Lloret , Alejandro Canovas, Joel J. P. C. Rodrigues and Kai Lin, A network algorithm for 3D/2D IPTV distribution using WiMAX and WLAN technologies, Multimedia Tools and Applications, Vol. 67, Issue 1, pp 7-30, November 2013.

[134] Jaime Lloret, Miguel Garcia, Jesus Tomás and Fernando 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.

[135] Miguel Garcia, Diana Bri, Fernando Boronat and Jaime Lloret, A new neighbour selection strategy for group-based wireless sensor networks, Fourth International Conference on Networking and Services (ICNS 2008), pp. 109-114, Gosier, Guadeloupe, March 16-21, 2008.

[136] Mohamed Lehsaini, Herve Guyennet and Mohammed Feham, Cluster-based Energy-efficient k-Coverage for Wireless Sensor Networks, Network protocols and Algorithms, Vol. 2, Issue 2, pp. 89-106, 2010.

[137] Jaime Lloret, Lei Shu, Raquel Lacuesta and Min Chen, User-Oriented and Service-Oriented Spontaneous Ad Hoc and Sensor Wireless Networks, Ad Hoc & Sensor Wireless Networks, Vol. 14, Issue 1-2, pp. 1-8, January 2012.

[138] Alejandro Canovas, Diana Bri, Sandra Sendra and Jaime Lloret, Vertical WLAN Handover Algorithm and Protocol to Improve the IPTV QoS of the End User, IEEE International Conference on Communications (ICC 2012), pp. 1901-1905, Ottawa (Canada), June 10–15, 2012.

[139] Congzhou Zhou and Nicholas Maxemchuk, Distributed Bottleneck Flow Control in Mobile Ad Hoc Networks, Network protocols and Algorithms, Vol. 3, Issue 1, pp. 22-45, 2011.

[140] R. C. Suganthe amd P. Balasubramanie, Improving QoS in Delay Tolerant Mobile Ad Hoc Network Using Multiple Message Ferries, Network protocols and Algorithms, Vol. 3, Issue 4, pp. 32-53, 2011.

[141] Lutful Karim, Alagan Anpalagan, Nidal Nasser and Jalal Almhana, Sensor-based M2M Agriculture Monitoring Systems for Developing Countries: State and Challenges, Network Protocols and Algorithms, Vol. 5, Issue 3, pp. 68-86, 2013.

[142] Miguel Edo, Alejandro Canovas, Miguel Garcia and Jaime Lloret, Providing VoIP and IPTV Services in WLANs, Chapter of the Book Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts, IGI Global, ISBN13: 9781609600426, pp. 426-444, 2011.

[143] Lyes Khoukhi and Soumaya Cherkaoui, Intelligent QoS management for multimedia services support in wireless mobile ad hoc networks, Computer Networks, Vol. 54, Issue 10, pp. 1692–1706, July 2010.

[144] Taner Çevik and Abdül Halim Zaim, A Multichannel Cross-Layer Architecture for Multimedia Sensor Networks, International Journal of Distributed Sensor Networks, Vol. 2013, Article ID 457045, 2013.

[145] Zhiyuan Li, Junlei Bi and Siguang Chen, Traffic Prediction-Based Fast Rerouting Algorithm for Wireless Multimedia Sensor Networks, International Journal of Distributed Sensor Networks, Vol. 2013, Article ID 176293, 2013.

[146] Jaime Lloret, Carlos Palau, Fernando Boronat and Jesus Tomas, Improving Networks Using Group-based Topologies, Computer Communications, Vol. 31, Issue 14, pp. 3438-3450, September 2008.