# Multimedia System for Emergency Services over TETRA-DVBT Networks

Román Belda, Ismael de Fez, Francisco Fraile, Víctor Murcia, Pau Arce, Juan Carlos Guerri
Institute of Telecommunication and Multimedia Applications
Valencia, Spain
E-mail: jcguerri@dcom.upv.es

**Abstract**

*This paper describes the main challenges of implementing a Multimedia on Demand system (TetraMoD) for emergency scenarios. The objective of this system is to offer a complete solution for providing multimedia services (Video o Demand and Reliability File Transfer) to a rescue team in an emergency situation. This multimedia system will improve the traditional emergency communications. We propose to integrate the use of different networks, TETRA and DVBT, characterized by a wide geographical coverage and broadband broadcast, respectively. This system consists of the following elements: a TetraMoD Gateway - the unit deployed on as interface between TETRA and DVBT networks- and a Middleware TetraMoD Client -the software for the client terminal. A real scenario and example of communication using standard protocols (SDP, RTP/RTCP, RTSP, FLUTE) are showed.*

## 1: Introduction

The possibility of accessing to up-to-date information, in real time and multiple formats (audio, video, images, text, binary files, etc) represents a key factor for the majority of communication systems. Specially important are emergency environments, disaster situations, medical attention, security and so on; where the communication between emergency teams such as police, fire-fighters, ambulances and the control headquarters or hospitals are crucial for making the right decisions at the right time.

Nowadays, most telecommunication systems targeting security and emergency response are based on the Terrestrial Trunked Radio (TETRA) network [1]. This standard is deployed worldwide, especially on Europe, and offers both support for voice communications and as a data channel for low speed transmissions (28.8 kbps per carrier). Nevertheless, despite its evolution (Release 2) so that it supports new data services, it won't be able to carry multimedia services such as real-time video, image downloading, file transfer, remote patient monitoring and so forth needed in next generation public safety systems.

Multiple solutions have been proposed based on satellite, GMTS/UMTS, WiMAX and other technologies in order to solve the problem consisting of sending multimedia information to emergency teams, some of them focused on ad hoc networks deployment in an emergency scenario. However, plus the diffusion capability, the adopted solution should take into account some other equally important aspects like coverage, scalability, security, QoS support, and low implementation cost.

Amongst the different works related to emergency environments and developed over the last years, we can mention the Wideband Wireless Local Area Network (WWLAN) Project [2], focused on the evaluation by simulation of heterogeneous satellite and WLAN network facilities, allowing audio-video communications between the fire brigade acting in a disaster area and the Control Centre that coordinates operations. Another project would be ADAMO [3], which is focused on how an ad-hoc wireless network can be securely connected and extended into a larger wireless network based upon technologies such as WIMAX or WIFI Mesh. The ADAMO project also investigates the integration of this environment with a TETRA network used by emergency services. Another work is WISECOM Project (Wireless Infrastructure over Satellite for Emergency COMmunications project) [4] focused on a description of the unit deployed on the disaster area. It is used as an interface between the selected satellite system and different wireless local access technologies (GSM, UMTS and TETRA over satellite).

However, there is not any solution based on integration between TETRA and DVB-T networks. DVB-T network [5] should be considered as an excellent candidate for broadcasting in view of features like wide coverage, multicast transmission capability, bandwidth, multimedia transmission, and low cost DVB-T receiver availability. The proposed solution in this paper is based on TETRA and DVB-T networks integration, which allows multimedia communications to improve the traditional emergency communications.

The rest of the paper is organized as follows: in Section II the considered scenario and protocols are described; in Section III a description of the middleware software deployed in the client terminal and the gateway implemented as interface between TETRA and multimedia server is provided. Next, the equipments used and the user interface programmed are showed. The paper concludes by looking at the conclusions and future work in Section V.

## 2: Scenario and Protocols

The goal of the TetraMoD system is the creation of a media on demand delivery platform, based on multimedia broadcasting over DVB-T networks and the use of the TETRA network as interaction channel. The considered scenario (Fig. 1) uses the two networks altogether: DVB-T network, with media broadcasting capabilities, used for transmitting the feeds and the TETRA network, used for requesting media presentations and transmission control. Both of these technologies are known for their ability to provide a high coverage as well as a high security degree for communications. Thus, this alternative is presented as an ideal solution for providing on demand downloading services deployment as an added value facility for TETRA communications. The contents offered through this platform are heterogeneous, that is, they can be both data files as media streams. The latter could be both streaming of stored presentations as well as live feeds.

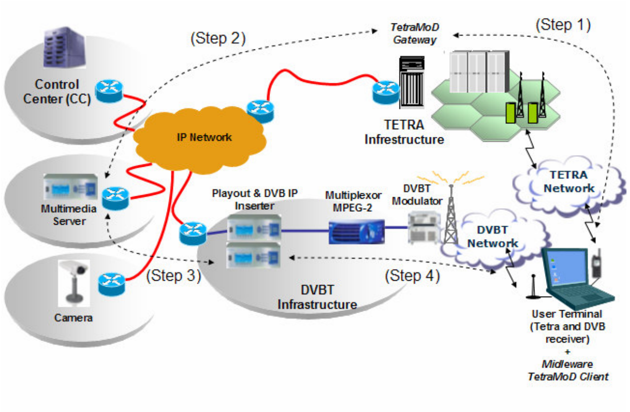


**Figure 1: Multimedia System Architecture**

A typical use case of this system is described next. Once an emergency call is received at the Control Centre, the research and operating units (fire-fighters, police, medical teams, etc) arrive and gain access to the disaster area. The rescue team has a terminal with access to TETRA and DVB-T networks, as well as the TetraMoD Client middleware installed on it. After inspecting the land and analysing the situation, it is crucial to possess the greater amount of information about the emergency area as possible in order to face the situation. Therefore, the rescue team will send a request to the Control Centre querying multimedia information about the emergency area-ranging from access routes, roads, buildings, maps, and so forth- using the TETRA network for this purpose so they can take advantage of its high coverage, dedicated bandwidth and secure access channel (Step 1). After that, the request is received and processed by the TetraMoD Gateway and redirected to the Content Server that actually provides the indicated content (Step 2). The needed multimedia information will be broadcasted over the DVB-T network and later received by the terminal client in order to improve making decisions (Steps 3 and 4).

The TetraMoD platform described is make up of several equipment elements, each one charged with a specific task in order to accomplish the steps needed to supply multimedia support for emergency teams. The IP Encapsulator (IPE from now on) has to gather the different media stream data coming from the Content Server and encapsulate them within elementary streams using the MPE protocol. The multiplexer combines a number of MPEG-2 elementary streams originated at different sources and arranges them within a single MPEG-2 Transport Stream. Last of all, the modulator generates a radiofrequency output signal for transmitting the desired Transport Stream in a given channel (Step 4).

On top of the platform architecture, the given solution uses IP protocols. In this manner, it provides a number of advantages like the following: a common layer to work with heterogeneous networks, interaction with currently available content delivering services based on IP as well as perspectives of imminent deployment over wireless networks. Also, the use of IP protocols is relevant regarding to resource management if we take into account that the available bandwidth for the service in the multiplex is limited, so use of the media on demand service by several users simultaneously can saturate the broadcast link. In this scenario, IP technology provides different mechanisms to optimize the bandwidth usage and control the QoS offered to the end user. The fact that neither DVB-T nor TETRA were originally conceived for carrying IP datagrams doesn't entail any inconvenience given that there have been defined standard mechanisms for data transmission over both of these technologies. A briefly description about the common protocols that have been used to develop this system follows.

- MPE (*Multi-Protocol Encapsulation*): A data link layer protocol defined by DVB which provides means to carry packet oriented protocols (like for instance IP) on top of MPEG-2 Transport Stream (TS). MPE uses MPEG-2 Private Table sections to carry the user datagrams.

- PEI (*Peripheral Equipment Interface*): A standard interface defined in TETRA for connecting a data terminal (e.g. a PC) to a TETRA radio terminal. It allows the data terminal equipment to use data transmission services over TETRA, like the transmission and reception of short data messages (SDS messages).

- SDP (*Session Description Protocol*) [6]: A standard protocol (RFC 2327) intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP does not provide the content of the media form itself but simply provides a negotiation between two end points to allow them to agree on a media type and format, for example during a RTSP session. It also reports connection information (IP addresses and ports), bandwidth information, etc. about the session being described.

## 2: Emergency Services

The system supports two kinds of services. On the one hand, the delivery of video feeds in real-time (video streaming). On the other hand, a file transfer service. An explanation of each one follows.

### 2.1: Video on Demand Service

With regard to the video streaming, the developed system allows the users of the client application to request both stored video presentations as live feeds. These videos are transmitted using the RTP protocol over a DVB-T channel.

In an emergency situation, when a member of a Public Safety and Disaster Recovery (PSDR) organization asks for a surveillance video camera transmission, every other member of the same emergency response team will be able to see the images. In this way, when a disaster like a traffic collision took place, the emergency units going to the affected area could see the traffic cameras located near that zone so they could have a visual perspective in order to conduct their tactical and emergency operations in a more precise manner.

The major protocols used with this service are:

- RTSP (*Real Time Streaming Protocol*): An application protocol [7] for use in streaming media systems which allows a client to remotely control a streaming media server, issuing VCR-like commands such as "PLAY" and "PAUSE", and allowing time-based access to stored media presentations and live feeds. It's worth noting that the sending of streaming data itself is not part of the RTSP protocol, but data is sent out of band using for instance the standards-based RTP as the transport protocol for the actual audio/video streaming. Therefore, the RTSP protocol acts somewhat as a metadata channel.

- RTP/RTCP (*Real Time Transport Protocol/Real Time Control Protocol*): Standard protocols to enable multimedia communications over a best-effort network without a resource guarantee or QoS. RTP/RTCP provides sequence numbering and time stamping of multimedia packets as well as a feedback channel to obtain reception statistics. This pair of protocols has become, nowadays, the standard for multimedia streaming over IP networks.

## 2.2: File Transfer Service

The system also provides file sending capabilities for any kind of file format. This way, reception of both images as text documents amongst others is possible. For this purpose FLUTE protocol is utilized. In our use case, for instance, the ambulance services can request a map of a secondary route where a traffic collision has taken place so they can rapidly access to the exact location.

Another clear example is information update performed by the police. An interesting application is the downloading of a robbed vehicles list on a daily basis so every day the police patrol can obtain an up-to-date registration numbers list using TETRA SDS requests. Since the file sending is multicast, the reception will be very fast and straightforward. The file can even be sent continuously so the downloading process doesn't require a previous request.

FLUTE (*File Delivery over Unidirectional Transport*) [8] is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. The protocol uses ALC (*Asynchronous Layered Coding*) [9], which is specifically designed to provide massive scalability using IP multicast. FLUTE was created to distribute files massively, offering reliability in the downloading process, and therefore it allows an asynchronous delivery of content to an unlimited number of receivers from one or various senders. ALC uses LCT (*Layered Coding Transport*) [10] to provide session management functionality, a congestion control block, as well as a FEC (*Forward Error Correction*) building block [11] to provide reliability.

A FLUTE session or file delivery session consists of a set of logically grouped ALC/LCT channels associated with a single source sending FLUTE packets. An ALC/LCT channel is defined by the combination of a sender and an address associated with the channel by the sender. A receiver joins to the channel to start receiving the data packets sent by the source, and leaves it when stops from receiving data packets or when wants to close the session.

Obviously, receivers must know the multicast addresses of the channels before they can join them and start receiving data packets. This and other relevant parameters of the multicast session are provided through out-of-band mechanisms, such as SDP, which has been mentioned previously.

One of the fundamental components FLUTE is based on is the File Delivery Table (FDT), which provides a means to describe various attributes associated with the files that are included within the file delivery session, for example the file location or its numeric identifier TOI. The FDT is delivered as FDT Instances, which are described in XML language, through FLUTE before the transmission of each file begins. Also, it is possible to send a general FDT at the beginning of the transmission which describes the properties of all session files.

In the shown scenario, the FDT allows checking in a quick manner whether a file requested by the client is found in the available files list maintained by the server.

When the server wants to transmit a FLUTE packet, it generates encoding symbols based on the object to be delivered using FEC codes and sends them in packets over the channels associated with the current session. The file segmentation process in encoding symbols is depicted in Figure 2.

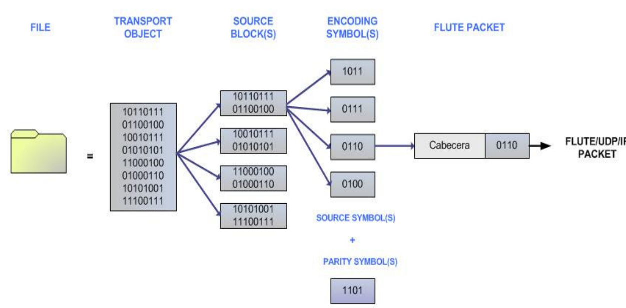


**Figure 2: FLUTE packet building process**

There are two kinds of sessions, File Delivery Sessions and File Carousels. The former consists on sending all the files associated with the session during a fixed time interval and then closing the session. In the latter, files are sent cyclically on a seamlessly ending loop. Thus, it provides a number of advantages, since the clients can connect to the session at any moment to receive any file. Furthermore, sessions can be static or dynamic (Fig. 3), depending on whether the contents of the session change during its lifetime.

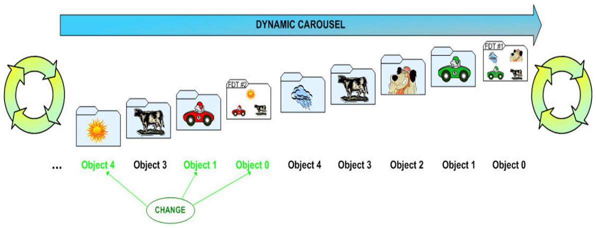


**Figure 3: Dynamic carousel example**

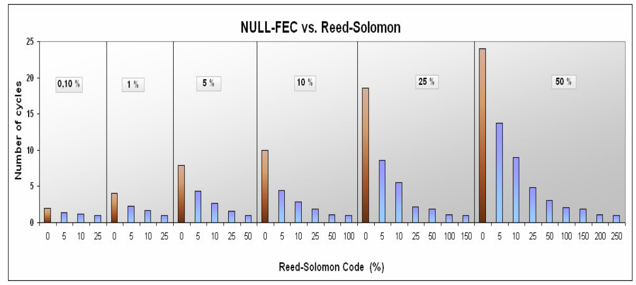


**Figure 4: Number of cycles vs. coding**

### 2.2.1: Grouping and Encoding

FLUTE protocol allows sending grouped files which have some common feature. Grouping files makes possible a clear and fast download for the client.

Grouping is performed by means of the File Delivery Table. The FDT holds every file property for the files that make up the carousel, such as their location, size, and so on. Also, it provides one or more fields in order to indicate the group each file belongs to.

In our scenario, wherever an emergency team is going to, it will be interested in gathering as much information about the target location as would be possible. Therefore, it can make a query related to that location that results in the server parsing and satisfying that request by sending all the files associated to the given group like maps, routes, and so forth. This way, a unique request will suffice in order to obtain all the needed information about the place.

On the other hand, one of the main features of FLUTE is the reliability that it provides for file downloading. This is achieved by means of two major mechanisms: on the one hand the continuous file forwarding using carousels and on the other hand the error control performed with FEC techniques.

As we saw in Figure 2, before a packet is sent the required redundancy is added. Unlike video on demand, the file transmission has implicit encoding thanks to FLUTE, so employing encoding blocks foreign to the architecture is not needed.

With the purpose of showing the advantages derived from use encoding, a number of tests have been carried out consisting on sending files with different redundancy through channels with different packet loss. For these tests Reed-Solomon encoding has been used. In Figure 4 we can see a comparative graphic that shows the needed number of transmissions for a given encoding type and redundancy employed. A redundancy value of zero corresponds to no encoding used. As we can note, the graphic shows clearly the convenience of using some type of error correction, mainly when the files are sent over a channel with a high packet loss ratio.

## 3: Implementation

An application targeting terminals like PCs with peripheral connectivity to TETRA and DVB-T networks has been developed with the purpose of validating the system proposed. TetraMoD Gateway provides a proxy service between the messages coming from clients with the TetraMoD middleware installed on and content servers. A description about TetraMoD Gateway and Client architectures, focused on implementation and functionality aspects, follows.

### 3.1: Middleware TetraMoD Client

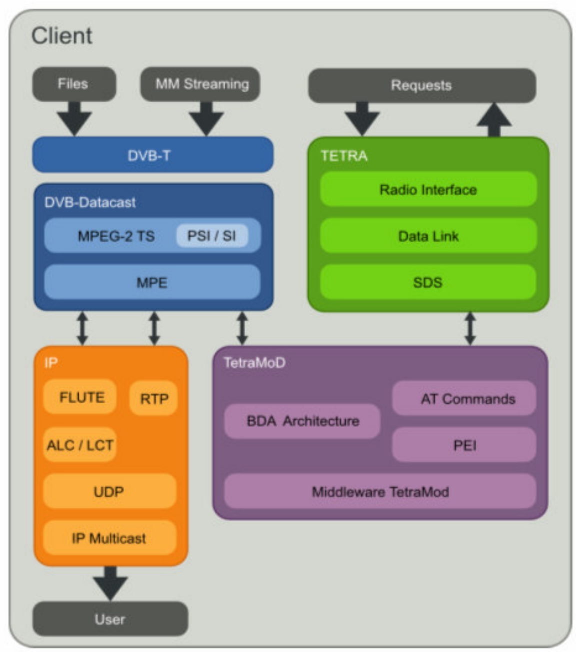


**Figure 5: TetraMoD Client**

The client middleware depicted in Fig.5 handles the media content queries as well as the DVB-T tuning, getting an IP flow encapsulated in a DVB-T channel as a result. These media queries are based on request/response messages. Since the communication channel to be used is the TETRA network, specifically the Short Data Service provided, these request/response pairs will be sent embedded within SDS messages.

The mentioned TetraMoD Client middleware has been programmed in Java. The reason for such a choice has been its ease of use for rapid prototyping of communication applications using TETRA devices via AT commands, thanks to the Java Communications API. The facilities provided by TETRA terminals are accessed through the standard TETRA Peripheral Equipment Interface (PEI), which standardises the connection of the radio terminal to an external device, and supports data transmission between applications resident in the device and the connected TETRA radio terminal.

On the other hand, the part of the system that performs the management, tuning and IP flow reception, has been developed using the TV and Broadcast Driver Architecture (BDA) API from Microsoft, thus programming in C++. The interaction between the BDA code and the Java middleware is done by means of a Java Native Interface (JNI) wrapper specially developed for this purpose.

The TetraMoD middleware permits requesting two kinds of services well differentiated: file transfer and audio/video streaming. File transfer is supported by FLUTE and ALC/LCT protocols, whereas media streaming is carried over the RTP protocol.

About this, the TetraMoD middleware incorporates the JVLC library from VideoLAN project, which allows media playback for many different formats and encodings.

### 3.2: TetraMoD Gateway

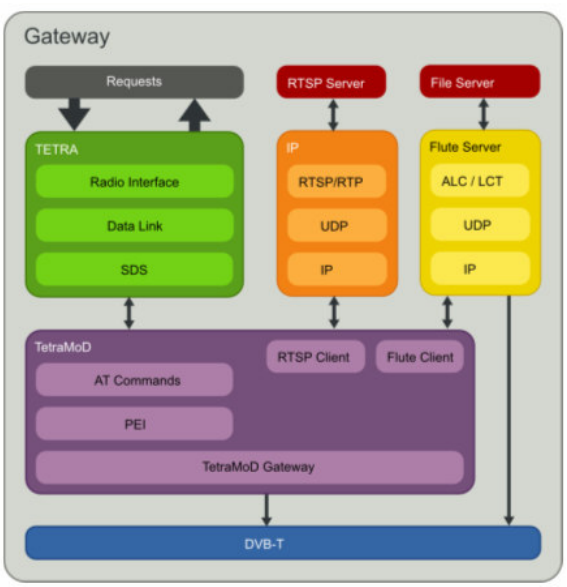


**Figure 6: TetraMoD Gateway**

TetraMoD Gateway (Fig. 6) has been programmed in Java, due to its cross-platform nature and straightforward use that applies for rapid prototyping of client and gateway applications compliant with network protocols as RTSP. Furthermore, the Java Communications API extension has been used to support both TETRA mobile stations configuration and control as well of the middleware layer implementation. Like in the client, this is done by making use of the PEI interface found on TETRA devices for serial communication between the mobile station and the terminal equipment (e.g. a PC). Content requests are embedded in SDS messages whose format will be different according to the content itself (video or files) as it can be seen in the next section.

In order to satisfy client requests the TetraMoD Gateway communicates with the Multimedia Server. As it is depicted in Fig. 6, the latter is in turn divided into two different servers, a RTSP Server for media streaming and a File Server for file transfer facilities. These servers will provide the media to be sent by the TetraMoD Gateway to the end user by means of DVB-T technology.

Also, the work of TetraMoD Gateway is to control the number of requests that are being satisfied at a given time and decide whether it is possible to accept new incoming client requests or not. In order to perform this task, TetraMoD Gateway parses the session description messages (using SDP protocol) sent from the Multimedia Server to find out the necessary bandwidth that must be allocated for each media stream and configure FLUTE server bit rate to accommodate state needs.

### 3.3: Description of end- to- end communication

*1) Video on Demand Service*

As it is shown in figure 7, communication between TetraMoD Gateway and the middleware layer is entirely performed through SDS messaging. These messages are processed by TetraMoD Gateway and then sent to the content server through normal RTSP over IP messages.

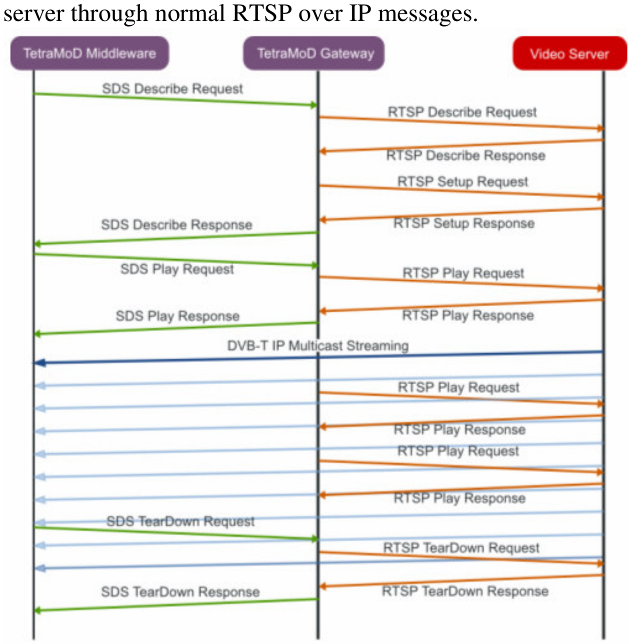


**Figure.7: Video on Demand end-to-end communication**

Tetra SDS messages have a limited body size of 140 characters so they could not be able to fit within a single SDS message. To address this issue it has been defined a header for the SDS data body that let us handle RTSP messages longer than 140 characters. This header has two fields of one byte each one. The former is used to identify if the SDS has an entire RTSP message, if the RTSP message has been split into several SDS messages or if it is the last SDS of a split RTSP message. The second field has information about the fragment number when the RTSP message needs more than one SDS. SDS Messages are based on RTSP ones but we have introduced some changes in order to improve the communication (that we have dubbed RTSP Lite). Changes are related to unnecessary data elimination and data field abbreviation.

**SDS Describe Request (using RTSP Lite)**
D rtsp://www.dcom.upv.es/video.mov/
CS: 1
ACC: app/sdp

**RTSP Describe Request**
DESCRIBE rtsp://www.dcom.upv.es/video.mov/ RTSP/1.0
CSeq: 1
Accept: application/sdp

\- - - - - - - - - - - - - - - - - - - - - - - - - - -

**RTSP Play Response**
RTSP/1.0 200 OK
CSeq: 4
Date: Wed, 27 Feb 2008 14:17:23 GMT
Range: npt=0-328

**SDS Play Response (using RTSP Lite)**
200 4
R: npt=0-328

**Figure 8: Examples of RTSP message and SDS message conversion**

Also, TetraMoD system has added a number of modifications to RTSP in order to minimize the number of SDS messages sent by the TETRA devices. One of them involves doing the transport initialization, i.e. the RTSP SETUP phase, without the intervention of the client. For this purpose, the RTSP DESCRIBE Response that TetraMoD Gateway sends to the client has been modified so that it contains the transport mechanisms information previously agreed with the Multimedia Server. The major benefit of this behaviour is the increased efficiency in terms of TETRA transport usage and the ability of handling the addressing space to be used in DVB-T MPE elementary streams.

RTSP servers are expecting the clients for keeping alive the RTSP session by means of RTCP feedback messages. As this is not possible in the present system, it will be TetraMoD Gateway that keep alive the media presentation 'on air' using additional RTSP PLAY Requests sent to the content server at regular intervals during the RTSP session life.

The following table summarizes the traffic saving achieved both for bytes transmitted as for number of SDS messages sent when implementing the conversion system previously described. As a result we get a 44% compression rate regarding message size and a 52% saving in number of messages.

**Table 1: Efficiency ratio of the message conversion**

| | | SDS Body Size (bytes) | | # SDS Messages | |
|---|---|---|---|---|---|
| | | RTSP | RTSP Lite | RTSP | RTSP Lite |
| **DESCRIBE** | **Request** | 85 | 57 | 1 | 1 |
| **DESCRIBE** | **Response** | 1047 | 637 | 8 | 5 |
| **SETUP** | **Request** | 108 x 2 | - | 1 x 2 | - |
| **SETUP** | **Response** | 151 x 2 | - | 2 x 2 | - |
| **PLAY** | **Request** | 93 | 72 | 1 | 1 |
| **PLAY** | **Response** | 81 | 21 | 1 | 1 |
| **TEARDOWN** | **Request** | 92 | 67 | 1 | 1 |
| **TEARDOWN** | **Response** | 24 | 7 | 1 | 1 |

| | RTSP | RTSP Lite | % |
|---|---|---|---|
| **Total Bytes** | 1940 | 861 | 44 |
| **Total SDS Messages** | 19 | 10 | 52 |

Nevertheless, the TETRA clients still will be able of closing the session by sending RTSP TEARDOWN Requests at any moment. Whenever the gateway receives live media streams requests for an ongoing RTSP session, it will do the negotiation process without querying the Video Server anymore. In this case, it will use the information obtained during the first media initialization phase to satisfy the subsequent client requests for the same media presentation. Moreover, the RTSP TEARDOWN Request sent by a given client will only be redirected to the Video Server if it's the last client interested in that media content, otherwise it will be discarded.

It's worth noting that the major bottleneck of the system is the available DVB-T channel bandwidth reserved for media streaming, so TetraMoD Gateway is perfectly capable and powerful enough for managing client connections without problems.

*2) File Transfer Service*

In the same way to the previous case, communication between user terminal and TetraMoD Gateway is performed using the Tetra SDS service, as it is shown in figure 9.

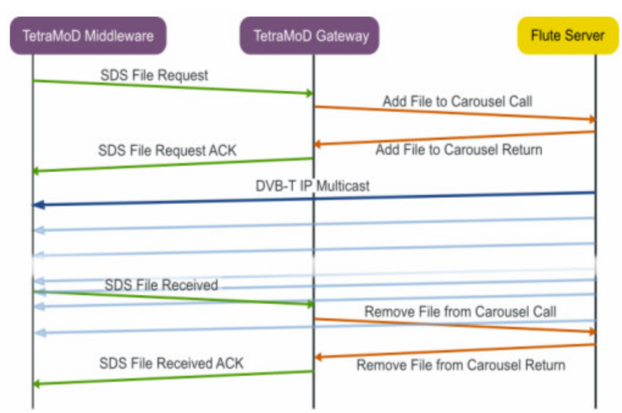


**Figure 9: File Transfer end-to-end communication**

The client sends a SDS File Request message querying the wanted file or file group. Then, the TetraMoD Gateway establishes a connection with the FLUTE Server requesting that the given file or file group be included within the file sending carousel. Hence, the FLUTE Server will transmit the carousel containing the file or group and all the clients who are tuned to that channel will be able of receiving the new content.

When the file has been successfully received, the client can send a message to the Gateway reporting this fact in order to the file could be removed from the carousel. The Gateway will check that the file is no longer expected by another client, and if this is the case, it will ask the File Server to modify the carousel.

As it is depicted in the figure, this model of Gateway-Server dialog doesn't need the Keep-alive messages to maintain the communication active.

## 3.4: User interface

In order to validate the system proposed it has been used the following equipment: a PC with a TETRA mobile station (EADS THR880i) attached to a serial port and a USB DVB-T Tuner (DIBcom DVB-T), an IP Encapsulator (SIDSA Vega IPDC), and a transmitter (ITIS V-XCAST DVB-T exciter). On the client side a PC application incorporates the middleware previously described.

The client application GUI (figures 10 and 11) has two tabs, each one corresponding to a different kind of content: video and file.

In the case of video streaming, the user can chose amongst the media available in the content list to the left of the application window. Once the media request has been satisfied, the video presentation will be shown by the media player on the right side (images, stored media presentations as well as live feeds).

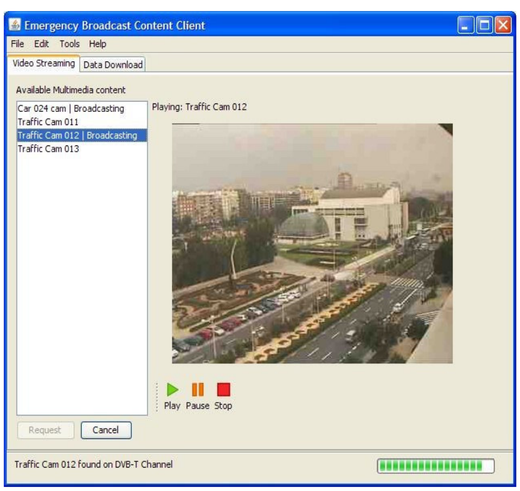


**Figure10: User application (Video Streaming Service)**

In the case of file transfer the process is analogous. The user just selects a single file or a complete group of them from the corresponding list in order to download them. This way, a thumbnail preview of the downloaded files will appear on the right pane of the application window.

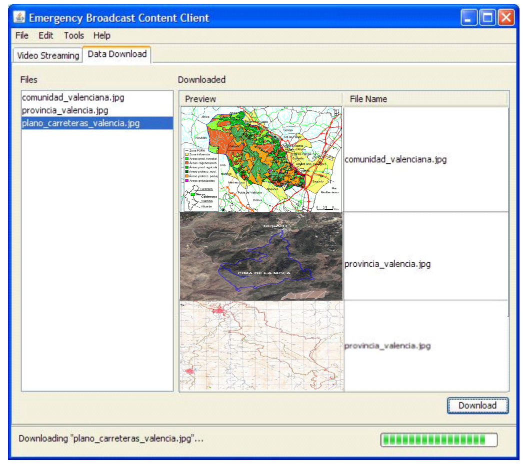


**Figure 11: User application (File Transfer Service)**

In both cases, upon selection, the corresponding SDS messages are generated and sent to the TetraMoD Gateway. Later on, once the client receives a number of SDS message containing the media description (IP address, ports, media type, encoding and format), it can request that the media streaming or file delivery starts. At the same time, the TetraMoD Client Middleware will tune the appropriate channel and extract the IP flow carried over.

If a video has been requested, the presentation playback will start and the image will be shown on screen, whereas if it is a file what has been requested a preview will be shown instead. The files will be saved in a directory previously defined by the application.

## 4: Conclusions

Today, emergency response teams need a secure, reliable and efficient communication system in order to carry out their tactical and emergency operations. TETRA networks are consolidated as a mature solution for Public Safety and Disaster Recovery agencies (like police and fire brigade) due to its reliability and confidentiality in the communication. Nevertheless, they are inefficient regarding to bandwidth provided, so a need for an added facility that satisfies this kind of services arises. The needed solution must provide high coverage, reliability and mainly high broadband capabilities. This way, it is possible to improve the service allowing both receptions of real-time video presentations as file downloading services in order to help to face the emergency situations.

Several solutions have been proposed in the last years. In this article we have opted to implement a solution based on the DVB-T standard, whose major features make it the more appropriate technology to provide broadband access to emergency response teams. The TetraMoD project has been successfully implemented emphasizing the efficiency of TETRA networks and the broadcast capabilities of DVB-T.

As future work, we present the need to enhance the security in the proposed platform communications. In this regard we propose the use of Raptor codes for communication reliability purposes. This type of encoding is recommended by DVB. Another line of future work is the development of client middleware for handheld devices, concretely for PDAs. These are a good option due to their continuous growth in memory, computational power and portability, being this last aspect very important in order to include the proposed system into body equipment.

## Acknowledgments

This work was founded by Abertis Telecom -Development of Technology Group (DETEC) and Mobile Communications Services Group (SECOM).

## References

[1] Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface (AI), European Telecommunications Standards Institute Std. EN 300 392-2, Rev. 2.5.2, Nov. 2005, work item reference: REN/TETRA-03129.

[2] O. Andrisano, A. Bazzi, A. Giorgetti, G. Pasolini, and V. Schena, "*Audio-Video Services for Emergency Scenarios over Heterogeneous Satellite-WiFi Networks*", in Proceedings of the IEEE Wireless Rural and Emergency Communications Conference (WRECON), 2007, Rome (Italy).

[3] IBBT. (2007, Jan.) Advanced disaster architecture with mobility optimizations. [Online]. Available: https://projects.ibbt.be/adamo/

[4] M. Berioli, N. Courville and M. Werner, "*Integrating Satellite and Terrestrial Technologies for Emergency Communications: the WISECOM Project*", in Proceedings of the Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), 2007, Vancouver (Canada).

[5] ETSI EN 300 744 – "DVB-T, Framing structure, channel coding and modulation for digital terrestrial television".

[6] M. Handley and V. Jacobson, "SDP: Session Description Protocol". RFC 2327 (1998)

[7] H. Schulzrinne, A. Rao and R. Lanphier, RFC 2326 "Real Time Streaming Protocol (RTSP)". RFC 2326 (1998)

[8] T. Paila, M. Luby, R. Lehtonen, V. Roca V and R. Walsh, "File Delivery over Unidirectional Transport (FLUTE)". RFC 3926 (2004)

[9] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo and J. Crowcroft, "Asynchronous Layered Coding (ALC) Protocol Instantiation". RFC 3450 (2002)

[10] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, M. Handley and J. Crowcroft, "Layered Coding Transport (LCT) Building Block". RFC 3451 (2002)

[11] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley and J. Crowcroft, "Forward Error Correction (FEC) Building Block". RFC 3452 (2002)