TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones

Revista Telemática. Vol.20 No.1, Enero-Marzo,2021, p.76-89 ISSN 1729-3804

DESIGN OF A RECEIVER OF MPEG TRANSPORT STREAM FOR ETHERNET NETWORKS

Ariel Catalá Valencia[1], Reinier Díaz Hernández[2]

[1] Technological University of Havana "José Antonio Echeverría" CUJAE, Calle 114 # 11901 / Ciclovía y Rotonda Marianao, La Habana, Cuba
[2] Institute of Research and Development of Telecommunications LACETEL, 34515 Km 14 1/2, Avenida Rancho Boyeros, La Habana,Cuba.
[1]e-mail: acatala@lacetel.cu
[2]e-mail: reinier@lacetel.cu

ABSTRACT

The design of a platform that receives MPEG streams through IP networks and is capable of retransmitting them is vital for the transmission of multimedia content over this type of networks, which are inherently heterogeneous. This work focuses on the selection of the most suitable mechanism for this, as well as on the processing of such streams in the designed platform. The streams processing contributes to mitigate the negative effects of heterogeneous networks on audio and video transmissions. This design is carried out in software using the Simulink tool. To validate the design, a transmission through an IP network of streams generated with the FFmpeg software is simulated. The results were obtained with the Wireshark tool and the Simulink logic analyzer. These results demonstrate that it is possible to perform such processing with the proposed design. However, its implementation in software is not efficient enough for its use due to the delays introduced. To solve this problem, the implementation of the current design on a hardware platform is proposed for future works.

KEYWORDS: MPEG, video broadcasting, Transport Stream, IP, Simulink

DISEÑO DE UN RECEPTOR DE FLUJO DE TRANSPORTE MPEG PARA REDES ETHERNET

RESUMEN

El diseño de una plataforma que reciba flujos MPEG a través de redes IP y sea capaz de retransmitirlos es vital para la transmisión de contenidos multimedia a través de este tipo de redes, las cuales son inherentemente heterogéneas. Este trabajo se enfoca en la selección del mecanismo más adecuado para ello, así como en el procesamiento de dichos flujos en la plataforma diseñada. Dicho procesamiento contribuye a paliar los efectos negativos de las redes heterogéneas sobre las transmisiones de audio y video. Dicho diseño se realiza de manera software con la utilización de la herramienta Simulink. Para la validación del diseño se simula una transmisión a través de una red IP de flujos generados con el software FFmpeg. Los resultados se obtuvieron con la herramienta Wireshark y el analizador lógico de Simulink. Dichos resultados permiten demostrar que es posible realizar tal procesamiento con el diseño propuesto. Sin embargo, su implementación en software no es suficientemente eficiente como para su utilización debido a los retrasos introducidos. Para resolver este problema se propone para futuros trabajos la implementación del presente diseño en una plataforma hardware.

PALABRAS CLAVES: MPEG, transmisión de video, Transport Stream, IP, Simulink

1. INTRODUCTION

TV broadcast networks

Digital Television (DTV) refers to the set of technologies for the transmission and reception of video and sound, through digital signals. In contrast to analog television, which encodes data in an analog manner, digital television encodes its signals in a binary manner, thus enabling the possibility of creating return paths between the consumer

Received: 2021-01-31, Approved: 2021-02-02
Sitio web:http://revistatelematica.cujae.edu.cu/index.php/tele

Revista Telemática. Vol.20 No.1, Enero- Marzo, 2020, p.76-89 ISSN 1729-3804

and the content producer, increasing the signal quality and the number of TV programs by decreasing the amount of radiofrequency spectrum used for TV broadcasts.

An important part of all digital television systems is the distribution network. It takes the information from the television studios and the content generation centers to the different transmitting centers that cover the entire territory where users are located. The distribution network can usually employ different means of communication, either by satellites, microwave links, or fiber optic networks, just to mention a few examples.

In Cuba, the most used way to connect to the main transmitting centers throughout the whole territory is ETECSA's national optical fiber. Therefore, it is not necessary to dedicate a link for the transmission of Digital TV contents, because they have multiplexed with the data that streams by this network, which constitutes an optimization in the network. These contents are transported using the MPEG transport streams (MPEG-TS), which are encapsulated on the RTCP-RTP/UDP/IP protocol stack. Consequently, it is possible to control the network status, improve the channel efficiency and manage the Quality of Experience (QoE). However, most of the used modulators do not directly support the IP input interface. This requires the use of additional equipment to adapt the received streams from the network. The adaptation allows the MPEG-TS streams, transported over the IP network, to be unpacked using the RTP (Real-time Transport Protocol) for later use in the receivers.

On the other hand, ***LACETEL*** is developing a DTMB modulator on a hardware platform based on FPGA (Field Programmable Gate Array) technology, which does not have the physical DVB-ASI interface commonly used in commercial modulators. Given this deficiency and, to validate the design of the modulator, an Ethernet interface was used, which is present in the available hardware and was implemented as an input interface to the modulator. This implementation has the necessary functionalities to receive the encapsulated Transport Stream (TS) directly over UDP/IP.

This article offers a solution to the lack of the DVB-ASI interface of the modulator by replacing it with an Ethernet input interface. For its design were studied different ways of implementation, which ended with a comparison in order to obtain the design with better performance.

In the following sections, it will be discussed the selection process of the transport mechanism, as well as the realization of the design that includes such transport mechanism and the results obtained with the realization of the design.

2. SELECTION OF THE TRANSPORT METHOD FOR MPEG TS

For the transmission of the video, it is necessary to compress it beforehand. This task is carried out using either of the two most used standards for this purpose: MPEG-2 or MPEG-4 (also called H.264/AVC). Both coding schemes are applied by the broadcast television transmission under the MPEG-2 packaging system, which is why these standards are so important [1].

Currently, there are three methods used for transporting such compressed video packets over IP, two of them use the RTP protocol.

TS over UDP

The first method, from now on called “direct TS over UDP”, selects some packets negotiated between the receiver and the transmitter and transports them as the payload of a UDP datagram as shown in Fig. 1. This method is described in [2, 3], but no specific reference is made to the number of TS packets that each datagram must carry.

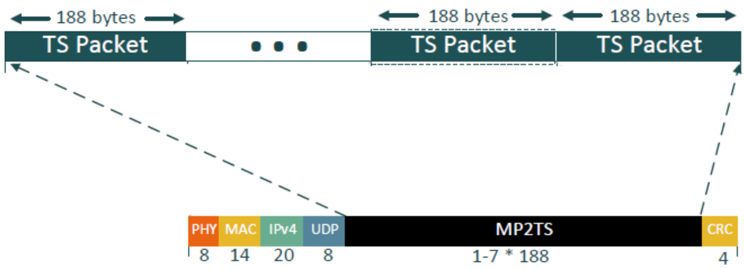

Figure 1: MPEG2-TS encapsulation directly on UDP.

The UDP protocol (User Datagram Protocol) is an interface that adds to the IP protocol (Internet Protocol) the ability to demultiplex the reception streams using a network abstraction called ports. It is a network transport protocol that allows IP datagrams to be sent without establishing a connection. Therefore, it does not establish flow control or congestion control, thus gaining speed when used as a transport protocol. On the other hand, the possibility of managing the retransmission after a bad reception is lost. However, its usefulness is wide in those services that are sensitive to delays because it offers reduced latency compared to the TCP protocol (Transmission Control Protocol) [4, 5].

This mechanism is supported by the framework StreamSim, proposed in [6]. Also, it was used for the transport of multimedia in experiments to measure the Quality of Service (QoS) of video transmissions on the Internet [7].

Due to the inherent characteristics of the UDP transport mechanism, this method does not provide a way to obtain information on the quality of reception. This is a disadvantage for content providers because they cannot diagnose when there are transmission losses and, therefore, it is difficult to know when there is a customer impairment. Besides, if used, all network devices (intermediate and final) need to support fragmentation because UDP implements packet fragmentation during transportation.

TS over RTP/UDP

The second method, from now on called “TS over RTP/UDP”, specified by the IETF in [2] and by the DVB-IPI group in [8] is designed for IP-based transport networks. It allows, using RTP, to carry an integral amount (7 for Ethernet-based networks) of MPEG-2 packets, which constitutes a MPEG-TS [9] as shown in Fig. 2. The use of the RTP protocol (Real-time Transport Protocol) provides the transport with reduced latency and works with the RTP header [5], which offers information about the content being transmitted and allows the QoS measurement as presented in [7, 10].

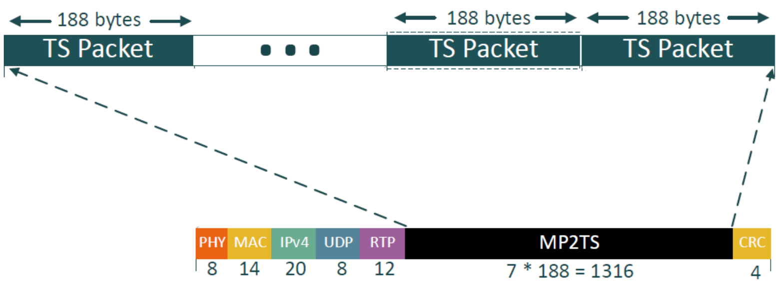

Figure 2: Encapsulation of MPEG2-TS over RTP.

Another advantage of the RTP protocol utilization is the use of RTCP (Real-time Transport Control Protocol), for periodically send control information associated with the data flow via UDP, as shown in Fig. 3. This information allows to add control and feedback capabilities to its delivery mechanism in a scalable way within large multicast networks and to offer a measurement of the Quality of Service. This is possible through the use of 4 basic functions: Feedback, Unique Identification, Bandwidth Control, and Minimum Negotiation of session information [8].

The StreamSim framework also supports this mode of transport [6]. Also, it is used in [11, 12] to validate a QoS monitoring system in IPTV networks. Furthermore, it has been selected for the transport of multimedia MPEG through wired transmission technologies [13] and on some IPTV platforms [14-18] . In summary, it is more widely used than other methods for transporting multimedia [19, 20].

Revista Telemática. Vol.20 No.1, Enero- Marzo, 2020, p.76-89 ISSN 1729-3804

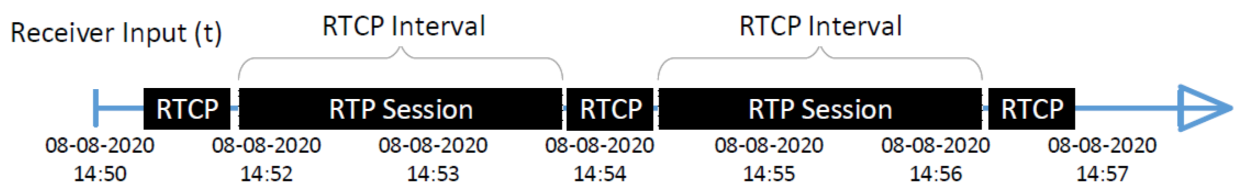

Figure 3: Receiving RTCP packets periodically during an RTP session.

Native RTP

The third method, from now on called "native RTP", makes up for the lack of knowledge of the type of packet being transported, unlike the methods described above [21]. It has been suggested in [22] to work only on MPEG-4/AVC (Advanced Video Codec) systems [9, 21, 22]. Like the previous method, it uses the RTCP protocol to maintain a data exchange on the status of the multimedia session.

This mode of transportation proposes a different method than those explained above. To achieve this, a separation of the audio and video streams into different RTP packets is accomplished through an abstraction called the MPEG access unit (AU) defined in [22]. This consists of the smallest data entity to be processed, for which synchronization information is assigned and stored as aligned octets, as shown in Fig. 4. In the case of audio, an access unit could be an audio frame, while for video, an image.

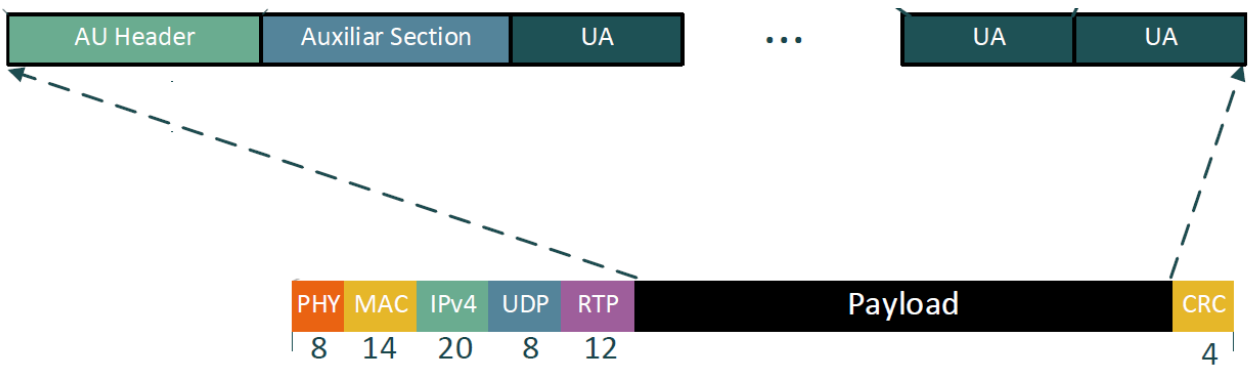

Figure 4: Native RTP packaging.

One of the strengths of this method is its resilience against errors, achieved by interleaving multiple access units when they correspond to the same content. However, this has a slight impact on the latency and complexity of the implementation [23].

This method is equally used to calibrate the system presented in [11]. Besides, it is employed to compare video streams for the service of IPTV in [24].

Comparison of methods for transporting MPEG over IP

For the selection of the method to be used, six aspects were analyzed. The first is the percentage represented by the payload header. Secondly, the transmission efficiency, given in equation (1), where k is the transmission efficiency, *PayloadSize* represents the size of the payload and *MTU* is the maximum transfer unit, which depends on the link. Besides, the complexity of the receivers and the adaptability they offer to the network were analyzed. Finally, the additional services it allows and the types of content it can transport were considered.

$$\boldsymbol{k} = \frac{\boldsymbol{PayloadSize}}{\boldsymbol{MTU}} \quad (\mathbf{1})$$

The results of the comparison are summarized in Table 1. As it is shown, the native RTP method is superior to the others in almost all aspects, but it presents a greater complexity in the receiver and the need for transcoding. On the other hand, the TS method over RTP/UDP allows, without much complexity, to obtain the benefits of the RTP header and the use of the RTCP protocol. For these reasons, the TS over RTP/UDP method was chosen.

Table 1. Comparison between methods for transporting MPEG flows

Aspect	TS over UDP method	TS over RTP/UDP method	Native RTP
Header	3,9 %	5.0 %	4,4 %
Transmission efficiency (k)	Between 0,125 and 0,88	0,88	0,93
Complexity of receivers and transmitters	Simple: It doesn't have to deal with the RTP header or RTCP packets.	Medium: It is necessary to deal with the RTP header, the RTCP packets and the scalable coding.	High. In addition to working with the RTP header and RTCP packets, it requires de-interleaving AUs.
Adaptation to network conditions	None. It does not maintain control over the state of transmission.	It is able to detect congestion with the use of RTCP packets.	It allows the sending of specific streams and the retransmission of those marked as crucial.
Added services	None.	The use of added services with this method leads to increased complexity and overutilization of the network.	It easily allows the transmission of specific flows to provide closed caption, multiple languages and extra camera angles.
Transport of MPEG 1,2 and 4	It is possible.	It is possible.	It is possible, although transcoding is required for MPEG 1 and MPEG 2 transport.

After selecting the transportation method, the Simulink tool of the Matlab software was chosen for the design of a platform to receive a MPEG transport stream. Moreover, the Wireshark traffic analyzer was chosen for monitoring the transport of content over the network and the FFmpeg software for transmitting content over IP.

3. DESIGN REALIZATION

In order to carry out the proposed design, the elements taken into account for a correct reception of the transport stream were analyzed. This reception can be guaranteed through several elements of the RTP header. Depending on its analysis, the encapsulated content is delivered or not to the receiver. Besides, it is necessary the implementation a scheme to guarantee synchronization, which can be affected by elements introduced by Ethernet networks such as delay, jitter, clock bias, and clock offset.

Evaluation environment

The design was done in Simulink, a Matlab tool for block programming, that allows the design and simulation of dynamic systems such as digital and analogical electronic systems, as well as network environments. These designs work in real-time and concurrently, contrary to traditional code-based programs, which run sequentially. This difference makes Simulink's environment able to generate environments that behave similarly to hardware and thus gain in processing speed.

TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol.20 No.1, Enero- Marzo, 2020, p.76-89 ISSN 1729-3804

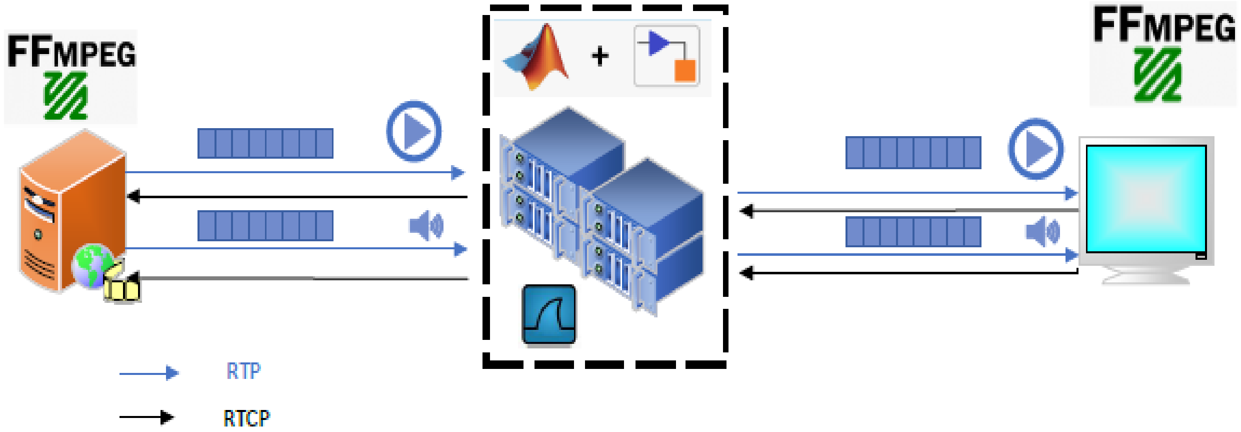

Figure 5: Evaluation environment to evaluate the performance of the receiver.

The environment in which the simulation was carried out constitutes a model of a real scenario, as shown in Fig. 5. The proposed design uses an input interface to simulate a reception and retransmits the content to verify the quality of the reception. The selected UDP reception ports were 3128 and 3129 for video streaming, while ports 4128 and 4129 were used for audio stream reception. The purpose of this test is to analyze the quality of the audio and video reception separately since both have different time requirements. The source of the stream in the presented scenario was a video, transmitted with the FFmpeg software, capable of sending a MPEG-TS stream over an Ethernet network. This stream was received in Simulink for the design and the processing of the packets. Additionally, the retransmission of the received content to another IP destination was implemented using the same transport mechanism as long as the stream meets a series of conditions. The retransmission is received by another instance of the FFmpeg software, which reproduces the content and allows the design performance to be analyzed in terms of reproduction quality.

Both instances of FFmpeg and the simulation were executed on the same computer. However, the use of network ports and IP addresses allows the simulation of a network architecture used by IP television distributors. Besides, the entire process is evaluated with the help of Matlab's logic analyzer to inspect the structure of the received and transmitted packets. The Wireshark traffic analyzer was used to obtain data related to the transport of the packets such as jitter and delay. Additionally, other network variables such as packet loss and network protocol usage were monitored. This analyzer supports MPEG TS frames, allowing an efficient inspection of the information with its associated fields.

Diagram of the receiver

The design was conceived as a receiver for the transport of MPEG streams using the TS over RTP/UDP method similar to the one presented in [25]. This design receives, through an IP interface, one or several MPEG transport streams encapsulated with the RTP protocol. This reception allows the analysis of the streams using the tools provided by the RTP header. The interface designed to receive MPEG streams over Ethernet networks fills the gap of the modulator designed in ***LACETEL***, which does not have an asynchronous serial interface (ASI), which is common in the transport of MPEG streams.

The scheme of the receiver was made using functional blocks to obtain a modular design easily modifiable. The parts in which it can be decomposed are Reception, Decomposition, Header Validation, Synchronization, Repackaging, and Transmission, as shown in Fig. 6. These stages are detailed below.

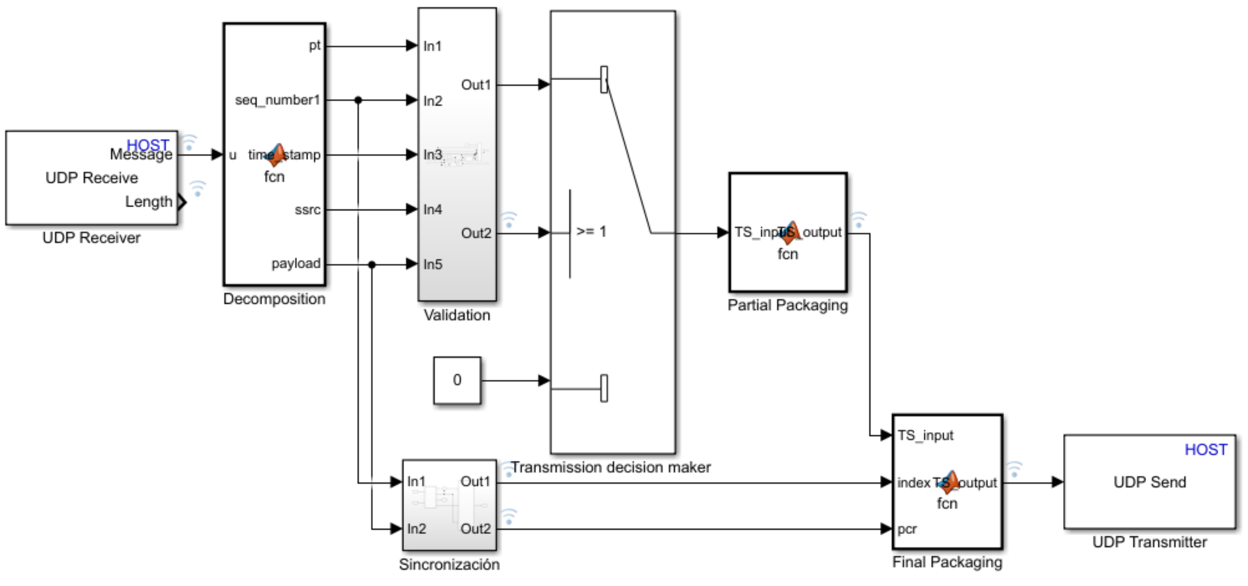

Figure 6: Simulink design of the presented receiver.

Reception and decomposition

To receive the stream, one of Simulink's blocks was used, allowing data to be received via an Ethernet interface. This was configured as a UDP port according to the proposed design because the reception of RTP packets is done over the UDP protocol. After the reception, the packet is broken down into simple elements, as shown in Fig. 7. and stored in RAM as is suggested in [25]. Each of these elements is needed independently to ensure that the receiver can implement the functions of the RTP protocol. This was necessary because of the lack of a block able to perform such decomposition in a versatile manner.

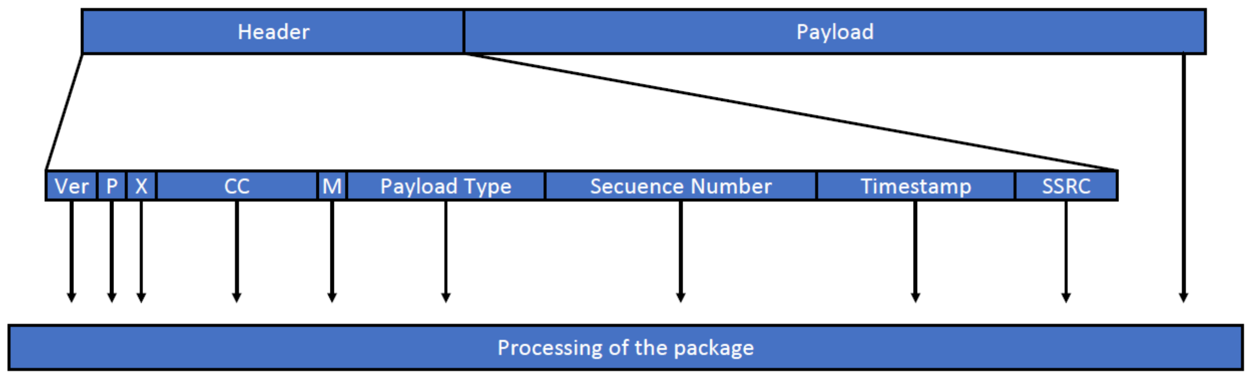

Figure 7: Decomposition of the package in the receiver.

Header processing

This stage of the design ensures that the reception of the packet complies with a set of rules that guarantees that it is a MPEG-2 stream transported via the TS over RTP/UDP method. This is essential for discarding packets from sources that are not using the presented method and therefore carry unexpected content [26]. To achieve this, an algorithm was designed to impose certain validation rules, as shown in Fig. 8. These rules ensure that version 2 is used, which is the most updated; that no padding is inserted into the packet, and that the type of payload is MPEG, which is marked with the number 33 according to [27, 28]. Besides, it verifies the synchronization source (SSRC) as the same as the content's source, which is equivalent to the absence of CSRC (Contribution Source). Besides, the SSRC of the stream must be equal to the SSRC of the desired program, thus avoiding the reception of unwanted content. Similar validations are performed during real-time reception of MPEG-2 streams on hardware systems [29].

TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol.20 No.1, Enero- Marzo, 2020, p.76-89 ISSN 1729-3804

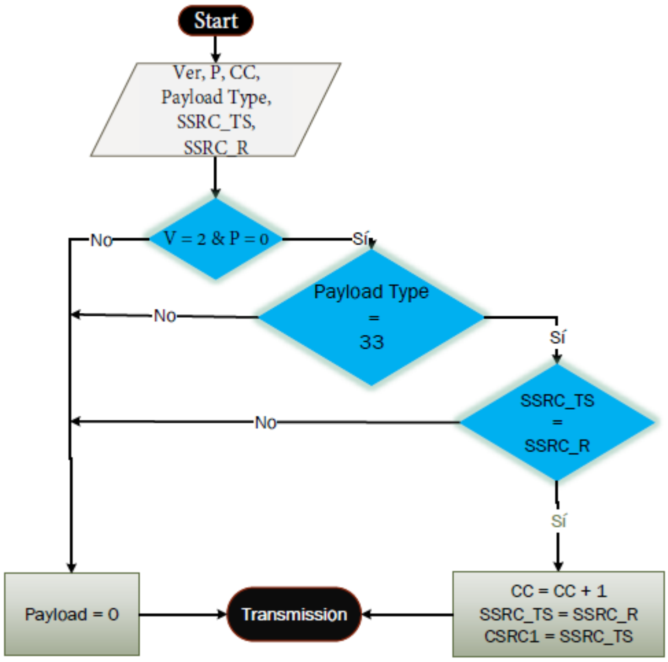

Figure 8: Algorithm for the validation of the received package.

If these rules are met, as shown in Fig. 8. three modifications are made in the header: the establishment of a new SSRC, the creation of the CSRC field with the source SSRC value, and the establishment of a CSRC in the CC field. The latter allows the receiver to recognize the existence of the CSRC, and the SSRC. Then, the modified header and payload are passed to the next stage of the design. Otherwise, a null vector is passed, which means that nothing is transmitted. The decision to transmit a null packet, rather than no packet at all, was made in the design since the transmission block offered by Simulink does not provide an interface for blocking transmission.

Synchronization

In digital television systems and heterogeneous networks, the packets carrying the TS must pass through several processing levels. Some of these are the addition or extraction of flags, tables (such as the PMT), and the padding, which usually cause a late output of the packet. There are also other processes such as checking, filtering, and adding headers during the path to the receiver. Those are typical of data networks and cause a delay in packet transport. Such delays lead to a time lag between the clocks of both systems, and thus to a necessary correction of the program clock reference (PCR). Another reason for changing the PCR of a packet is the insertion frequency which varies according to the standard used. The MPEG-2 standard uses a period of 100 ms while the DVB standard uses a minimum period of 38 ms [30, 31].

The PCR is a 42-bit field located in the adaptation field of the TS. It consists of a 9-bit part (PCR_{ext}) that increases at a rate of 27 MHz and is truncated at 300 and a 33-bit part (PCR_{base}) that increases at a rate of 90 kHz, as shown in equation (2), in which the PCR of packet *i* is analyzed. Both work like 2 cascaded counters where each time the first one reaches its final state, the second one increases [32, 33].

$$PCR(i) = PCR_{base}(i) * 300 + PCR_{ext}(i) \tag{2}$$

The *PCR* correction of a transport stream relies on equations (3) and (4), where *PCR* is the value of the incoming *PCR*, *PCR'*, the value of the corrected *PCR* and Δ *PCR*, a compensation constant defined by the difference between the current packet delay ($\Delta\, PCR_{act}$) and the average packet delay ($\Delta\, PCR_{const}$).

$$PCR' = PCR + \Delta PCR \tag{3}$$
$$\Delta PCR = \Delta PCR_{act} - \Delta PCR_{const} \tag{4}$$

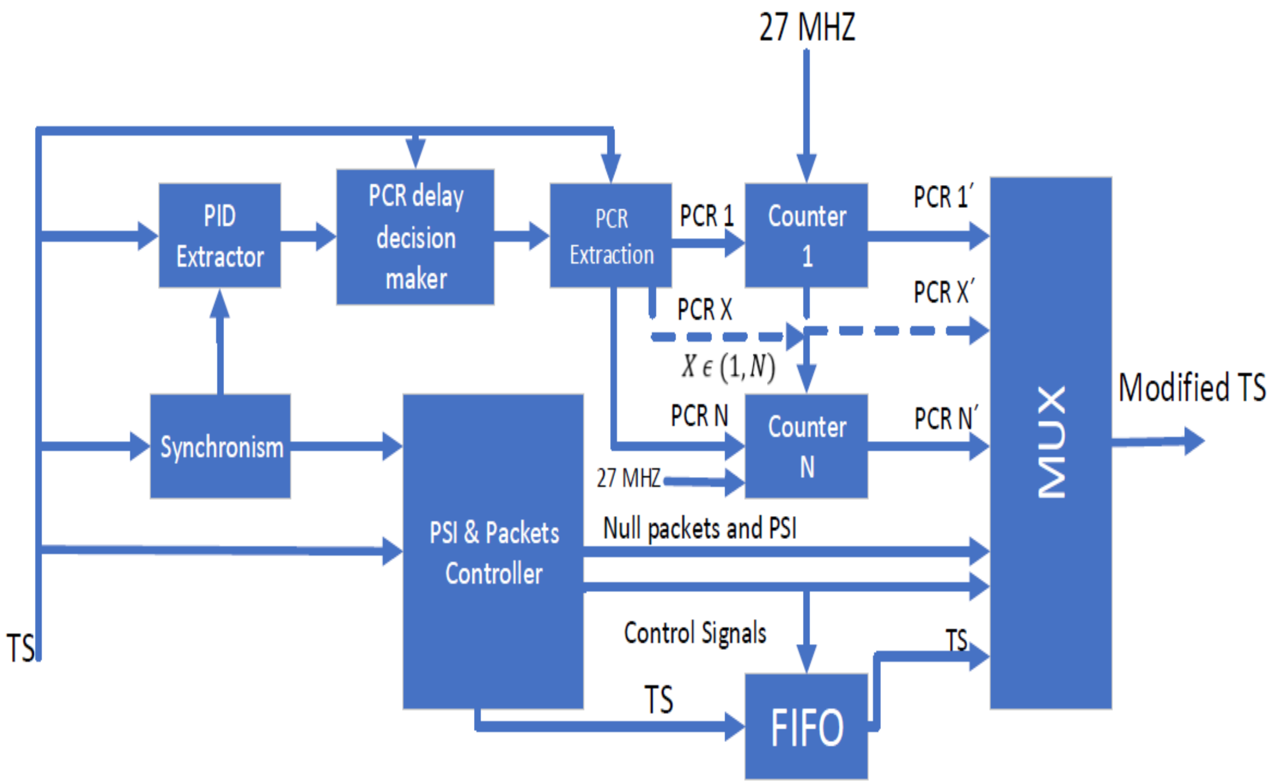

Figure 9: Scheme used for PCR correction.

There are several schemes for the correction of PCR [34]. In the present paper, it has been chosen the dedicated PCR-counters method is the easiest correction scheme to implement. The implementation of this correction system needs a counter for each program restarted with the arrival of each package. Furthermore, a LUT (Lookup table) is needed to map each PCR with each program. This can be a bad optimization of the system's resources in case of an increase in the number of programs [32]. The design of this correction scheme is presented in Fig. 9.

For the replacement of the PCR, it is necessary to identify those packages where it is located. This is achieved with an algorithm to detect their presence through the adaptation field and indicate their position in the TS packet, as shown in Fig. 10.

The counter used for PCR correction has a limitation due to the time granularity of the personal computer processor. Such granularity is in the order of milliseconds, however, a much higher frequency is needed [35], because of the systems use 27 MHz timers to generate the PCR.

TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol.20 No.1, Enero- Marzo, 2020, p.76-89 ISSN 1729-3804

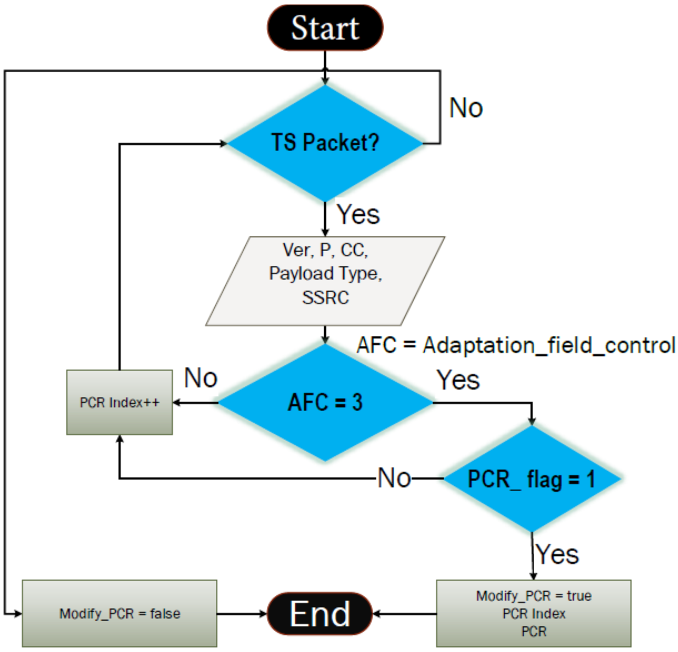

Figure 10. Algorithm for the identification of the PCR and the index corresponding to the package in which it is found.

Repackaging and transmission

In this stage, the processed header is jointed with the packet payload, where the PCR is replaced when present. This allows an RTP packet to be formed if the conditions are met, otherwise, a null packet is generated. The structured packet is then transmitted to another IP address via the UDP interface using the RTP protocol of the Simulink model. An interface for the design evaluation is located at this retransmission address by reproducing the stream using FFmpeg software.

4. RESULTS

Design validation

For the validation of the proposed design, the test scenario described above was assembled. Three MPEG video streams with different sizes (352x240, 720 x 480, and 1280 x 720) and one audio stream were generated with the FFmpeg software. These streams were transported with the RTP protocol and received in the IP interface. The generated network traffic was analyzed with the use of Wireshark. The analysis verified that the received packets complied with the rules designed in the discrimination algorithm described above. It was also verified that the packet exchange was performed with the RTP and RTCP protocols. Moreover, the Simulink logic analyzer was used to evaluate if the design performed the header processing and the PCR modification. Finally, the quality of playback was evaluated through its visualization.

Achieved results

During the reception, it was checked that RTP packets can be received on UDP port 3128, shown in Fig. 11. In this figure, it can be seen that the communication starts with the sending of an RTCP Sender Report packet to port 3129, as described in [8].

	Time	Source	Destination	Protocol	Length	Puerto	Info
1	0.000000	127.0.0.1	127.0.0.1	RTCP	88	65141,3129	Sender Report
2	0.000123	127.0.0.1	127.0.0.1	MPEG TS	1388	65140,3128	PT=MPEG-II transport streams, SSRC=0xC38937F7, Seq=3663, Time=2462805199 Service
3	0.000189	127.0.0.1	127.0.0.1	MPEG TS	1388	65140,3128	[MP2T fragment of a reassembled packet]
4	0.000230	127.0.0.1	127.0.0.1	MPEG TS	1388	65140,3128	[MP2T fragment of a reassembled packet]
5	0.000274	127.0.0.1	127.0.0.1	MPEG TS	1388	65140,3128	[MP2T fragment of a reassembled packet]
6	0.000300	127.0.0.1	127.0.0.1	MPEG TS	1388	65140,3128	[MP2T fragment of a reassembled packet]
7	0.001455	127.0.0.1	127.0.0.1	MPEG TS	1388	65140,3128	video-stream [MP2T fragment of a reassembled packet]
9	0.058494	127.0.0.1	127.0.0.1	MPEG TS	1388	65140,3128	[MP2T fragment of a reassembled packet] [MP2T fragment of a reassembled packet]
10	0.058917	127.0.0.1	127.0.0.1	MPEG PES	1388	65140,3128	[MP2T fragment of a reassembled packet] Program Association Table (PAT) Program
18	0.160068	127.0.0.1	127.0.0.1	MPEG PES	1388	65140,3128	[MP2T fragment of a reassembled packet] [MP2T fragment of a reassembled packet]

Figure 11: Receiving MPEG stream on port 3128 and RTCP packets on port 3129.

When analyzing the structure of a received RTP packet we see that the header complies with the required version (version 2), does not contain CSRC, and is composed of TS packets (ISO/IEC 13818-1). All these specifications can be seen in Fig. 12.

```
> Frame 5: 1388 bytes on wire (11104 bits), 1360 bytes captured (10880 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 65140, Dst Port: 3128
v Real-Time Transport Protocol
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: MPEG-II transport streams (33)
    Sequence number: 3666
    Timestamp: 2462805199
    Synchronization Source identifier: 0xc38937f7 (3280549879)
v ISO/IEC 13818-1 PID=0x100 CC=2
  v Header: 0x47010012
      0100 0111 .... .... .... .... .... .... = Sync Byte: Correct (0x47)
      .... .... 0... .... .... .... .... .... = Transport Error Indicator: 0
      .... .... .0.. .... .... .... .... .... = Payload Unit Start Indicator: 0
      .... .... ..0. .... .... .... .... .... = Transport Priority: 0
      .... .... ...0 0001 0000 0000 .... .... = PID: Unknown (0x0100)
      .... .... .... .... .... .... 00.. .... = Transport Scrambling Control: Not scrambled (0x0)
      .... .... .... .... .... .... ..01 .... = Adaptation Field Control: Payload only (0x1)
      .... .... .... .... .... .... .... 0010 = Continuity Counter: 2
    [MPEG2 PCR Analysis]
  Reassembled in: 7
> ISO/IEC 13818-1 PID=0x100 CC=3
```

Figure 12: Structure of a received RTP packet.

The results with the Simulink logic analyzer showed that, in the packets where the existence of the PCR was detected, this field was modified, which can be seen in Fig. 13. Besides, the design identifies the index of the TS packet with the PCR and the modification of the RTP header.

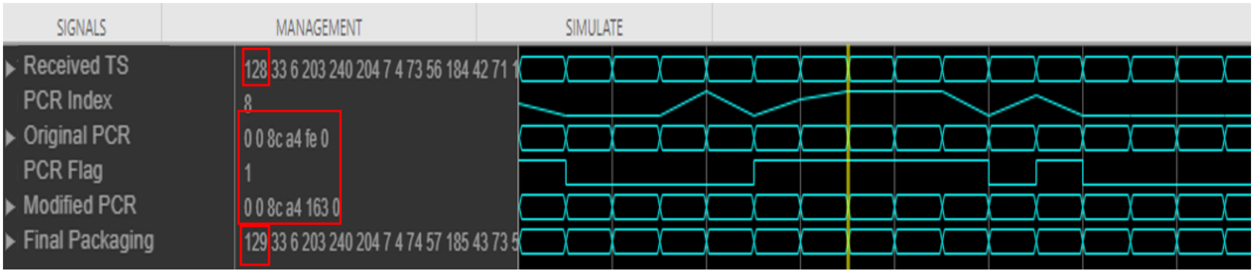

Figure 13: Processing a package with PCR.

The results of the retransmission with the Wireshark showed that both sessions have a close average jitter, as shown in Fig. 14. Furthermore, it is observed that each transmission has a different SSRC, since the design generates its own SSRC.

Source Address	Source Port	Destination Address	Destination Port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter	Mean Jitter
127.0.0.1	52478	127.0.0.1	4000	0x9020d0c5	MPEG-II streams	843	0 (0.0%)	649.156	57.066	11.496
127.0.0.1	52479	127.0.0.1	3128	0x8f1fcfc4	MPEG-II streams	844	0 (0.0%)	641.015	56.343	10.935

Figure 14: Statistics of both transmissions.

In order to check the quality of the processing, three streams with different resolutions were received and the quality of the reproduction was evaluated. Such procedure showed that the quality is inversely proportional to the size of the reproduction since the errors in the decoding process increased as the resolution did. Such errors in reception are due to synchronization problems resulting from performing the processing on a general-purpose computer.

Conclusions

The software design of the receiver in this article allows the reception of a MPEG transport stream over Ethernet networks. During the validation tests, it was demonstrated that operations using the RTP header allowed the analysis of the received content, which is useful to avoid the reproduction of unwanted content. However, the strong time constraints imposed by video transmission make it impossible to implement all the necessary processing in real-time. Another limitation is the granularity in the clock of general-purpose computers, which does not allow generating a timing similar to that of MPEG modulators. However, the present design makes up for the lack of an ASI interface in the FPGA-based modulator used by ***LACETEL*** to receive MPEG transport streams. Therefore, it is recommended the implementation of the presented design in that modulator and the evaluation of its performance.

ACKNOWLEDGEMENTS

The authors would like to thank ***LACETEL*** researchers, who made this research possible and helped to obtain the results. They also want to thank the Stack Over Flow community for the documentation about the FFmpeg software

REFERENCES

[1] L. B. Yuste and M. Montagud, "Understanding Timelines within MPEG Standards," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. XX, 2015.
[2] T. ETSI, "102 034: Digital Video Broadcasting (DVB)," *Transport of MPEG-2 TS based DVB services over IP based networks*, 2007.
[3] *ITU-T Recommendation: Full service VDSL - System architecture and customer premises equipment*, ITU-T, 2003.
[4] L. P. Larry, *Computer Networks: A systems approach*. Morgan Kaufmann, 2020.
[5] S. Afzal, V. Testoni, C. E. Rothenberg, P. Kolan, and I. Bouazizi, "A Holistic Survey of Wireless Multipath Video Streaming," *arXiv preprint arXiv:1906.06184*, 2019.
[6] A. M. Dethof, W. Robitza, and M.-N. Garcia, "StreamSim: A Video Streaming Simulation Toolchain for Unreliable Transport Mechanisms," in *de QoMEX–Eighth International Conference on Quality of Multimedia Experience, Lisboa*, 2016.
[7] M. Chiroiu, V. Şerbu, D. Rosner, and R. Rughiniş, "Analysis of video traffic quality over internet," in *2016 15th RoEduNet Conference: Networking in Education and Research*, 2016, pp. 1-6: IEEE.
[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC 3550: RTP: A transport protocol for real-time applications," ed: rfc 1889, January, 1996.
[9] T. U. Stefan Paulsen, Krzysztof Nowicki, "MPEG-4/AVC versus MPEG-2 in IPTV," *Proceedings of the International Conference on Signal Processing and Multimedia Applications and Wireless Information Networks and Systems*, 2012.
[10] M. Ivoševiæ, M. Vranješ, V. Pekoviű, and Z. Kaprocki, "Client-side solution for QoS measurement of video content delivery over IP networks," in *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, 2018, pp. 1-6: IEEE.
[11] A. Mozo Robles, "Diseño e implementación de un sistema de medición y monitorización de la calidad de servicio y experiencia de redes IPTV basadas en el códec H. 264," 2016.
[12] J. C. Cuéllar Quiñonez, "Modelo para la medición de calidad de la experiencia para el servicio de IPTV," 2018.

[13] Y. Endo, A. Hashimoto, M. Yamamoto, and T. Kurakake, "Latest on R&D on Wired Transmission Technologies," *Broadcast Technology*, 2016.
[14] G. Brito and A. Borja, "Plataforma de IPTV utilizando tecnología GPON para el servicio de video por suscripción de la CNT EP en la zona de cobertura de la central Izamba del cantón Ambato," 2017.
[15] J. Frnda, J. Nedoma, J. Vanus, and R. Martinek, "A hybrid QoS-QoE estimation system for IPTV service," *Electronics*, vol. 8, no. 5, p. 585, 2019.
[16] N. C. Casco Brito, "Simulación y evaluación de una arquitectura IPTV de video en tiempo real, mediante software libre," Escuela Superior Politécnica de Chimborazo, 2019.
[17] N. Goran, A. Begović, and N. Škaljo, "Comparing simulation model for objective QoE video evaluation with real IPTV test scenario during appearance of packet losses," in *2019 27th Telecommunications Forum (TELFOR)*, 2019, pp. 1-4: IEEE.
[18] A. Ortiz Alegre, "Despliegue de un sistema de IPTV en un emulador de redes," 2017.
[19] B. Shin, J. Abdullayev, and D. Lee, "An Efficient MAC Layer Packet Fragmentation Scheme with Priority Queuing for Real-Time Video Streaming," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, 2016, pp. 69-77: IEEE.
[20] C. R. Yánez Morales, "Diseño y simulación de una red de transporte de video en tiempo real en IPV4 mediante protocolo Unicast, dirigido a empresas de post-producción/edición de contenido multimedia con calidad de servicio," PUCE, 2017.
[21] A. MacAulay, "IP Streaming of MPEG-4: Native RTP vs MPEG-2 Transport Stream, ," *Envivio Whitepaper*, p. 3, Oct. 2005.
[22] "RFC 3640: RTP Payload Format for Transport of MPEG-4 Elementary Streams."
[23] *RFC 3640: RTP Payload Format for Transport of MPEG-4 Elementary Streams*, 2003.
[24] T. Uhl, J. H. Klink, K. Nowicki, and C. Hoppe, "Comparison study of H. 264/AVC, H. 265/HEVC and VP9-coded video streams for the service IPTV," in *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2018, pp. 1-6: IEEE.
[25] N. X. Shasha Yua, "Design and Implementation of TS over IP Gateway System," presented at the 8th International Congress of Information and Communication Technology (ICICT-2018), 2018.
[26] K. Kim, J. Kim, W. Zia, and L. Hyeonjae, "Method for receiving media and device thereof," ed: Google Patents, 2017.
[27] H. Schulzrinne and S. Casner, "RFC3551: RTP Profile for Audio and Video Conferences with Minimal Control," ed: RFC Editor, 2003.
[28] "RFC 2250: RTP Payload Format for MPEG1/MPEG2 Video."
[29] S. Zhang, X. Jin, J. Liu, and S. Jin, "Real-time Extraction of MPEG-2 Stream Based on FPGA," *DEStech Transactions on Computer Science and Engineering*, no. iceiti, 2017.
[30] M. Machmerth and C. Stoerte, "Accumulator Based PCR Restamping used for high Data Rate Transport Stream Transmission Adapters for A-VSB," *The 13th IEEE International Symposium on Consumer Electronics*, 2009.
[31] *ISO/IEC 13818-1:2000 Information technology — Generic coding of moving pictures and associated audio information: Systems*, 2000.
[32] B. C. d. Farias, E. B. d. L. Filho, and E. A. Bezerra, "A Proposal for Clock Reference Correction in MPEG-2 Transport Stream Processors," 2014.
[33] L. B. Yuste, "Time and timing within MPEG standards," in *MediaSync*: Springer, 2018, pp. 411-450.
[34] H. J. Savino and E. B. de Lima Filho, "Program clock reference correction in transport stream processors with rate adaptation," *Multimedia Tools and Applications*, vol. 76, no. 12, pp. 14107-14128, 2017.
[35] J. Lapierre, "A Practical IP Software Encode/Decode Implementation," presented at the SMPTE 2022-6,

ABOUT THE AUTHORS

ARIEL CATALÁ VALENCIA: Telecommunications and Electronics Engineer from the Universidad Tecnológica de La Habana "José Antonio Echeverría" CUJAE where he was an assistant student in subjects such as Digital Electronics I and II, Probability and Statistics and Digital Signal Processing. He has developed his work practices at the Institute of Research and Development of Telecommunications LACETEL, where he currently works as a scientific reserve researcher. His research interests are Digital Television Systems, Digital Signal Processing, Data Science, Machine Learning, and Digital Electronic Systems. ORCID: https://orcid.org/0000-0003-1671-2371

REINIER DÍAZ HERNÁNDEZ: MsC, Telecommunications and Electronics Engineer graduated in 2007 from the Technological University of Havana (CUJAE). He has been working as a researcher at LACETEL, Havana, Cuba, since October 2009; in August 2016 he obtained the category of "Researcher Aggregate". In February 2017 he concludes his Master's thesis and obtains the category that accredits him as Master of Science (MsC). The main

TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol.20 No.1, Enero- Marzo, 2020, p.76-89 ISSN 1729-3804

research topics he develops are related to digital communications and digital signal processing in broadcasting systems. ORCID: http://orcid.org/0000-0001-9439-4714

CONFLICT OF INTEREST

The author declares no conflict of interests.

AUTHOR CONTRIBUTIONS

- **Autor 1**: He developed the presented idea, verified the theoretical methods, designed, carried out the experiments, and wrote the manuscript. **(70%)**
- **Autor 2**: He conceived the original idea, discussed the results and contributed to the final manuscript. **(30%)**