SOFTWARE TESTING, VERIFICATION AND RELIABILITY
Softw. Test. Verif. Reliab. 2021; **00**:1–35
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/stvr

A Fuzzing-Based Test-Creation Approach for Evaluating Digital TV Receivers via Transport Streams

Fabricio Izumi[1], Eddie Filho[1,3], Lucas C. Cordeiro[2,3], Orlewilson Maia[1], Romulo Fabricio[1], Bruno Farias[1] and Aguinaldo Silva[1]

[1] *TPV Technology, Manaus, Brazil*
[2] *University of Manchester, Manchester, United Kingdom*
[3] *Federal University of Amazonas, Manaus, Brazil*

SUMMARY

Digital TV (DTV) receivers are usually submitted to testing systems for conformity and robustness assessment, and their approval implies correct operation under a given DTV specification and protocol. However, many broadcasters inadvertently misconfigure their devices and transmit the wrong information concerning data structures and protocol format. Since most receivers were not designed to operate under such conditions, malfunction and incorrect behavior may be noticed, often recognized as field problems, thus compromising a given system's operation. Moreover, the way those problems are usually introduced in DTV signals presents some randomness, but with known restrictions given by the underlying transport protocols used in DTV systems, which resembles fuzzing techniques. Indeed, everything may happen since any deviation can incur problems, depending on each specific implementation. This error scenario is addressed here, and a novel receiver robustness evaluation methodology based on non-compliance tests using grammar-based guided fuzzing is proposed. In particular, devices are submitted to unforeseen conditions and incorrect configuration. They are created with guided fuzzing based on real problems, protocol structure, and system architecture to provide resources for handling them, thus ensuring correct operation. Experiments using such a fuzzing scheme have shown its efficacy and provided opportunities to improve robustness regarding commercial DTV platforms. Copyright © 2021 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Digital TV, Fuzzing, Robustness Testing, Transport Stream, Testing Methodology.

1. INTRODUCTION

Over the last decades, many countries migrated from analog TV systems to digital TV (DTV) standards [1]. In Europe, *Séquentiel couleur à mémoire* (SECAM) and phase alternating line (PAL) [2] were replaced by digital video broadcasting terrestrial (DVB-T) [3] and, more recently, DVB-T second-generation (DVB-T2) [4, 5]. North America moved from national television system committee (NTSC) [2] to advanced television system committee (ATSC) [6], now being updated to ATSC 3.0 [7]. Japan adopted the standard named integrated services digital broadcasting - terrestrial (ISDB-T) [8]. In addition, most of South America chose its variant known as ISDB-T version B (ISDB-T$_B$) or Brazilian digital television system (SBTVD) [9], as a substitute to PAL, SECAM, and NTSC. Consequently, receivers in those regions should conform to specific DTV standards, usually accomplished with commercial testing systems or proprietary approaches.

[1] Correspondence to: eddie.filho@tpv-tech.com

Besides, DTV standards often include middleware specifications [10]. DVB-T is traditionally associated with multimedia home platform (MHP) and, more recently, hybrid broadcast broadband TV (HbbTV) [9]. At the same time, ATSC developed the advanced common application platform (ACAP) [11] and later released its ATSC 3.0 interactive content standard [12]. ISDB-T initially included the broadcast markup language (BML) standard and, currently, promotes Hybridcast [9]. Moreover, MHP and HbbTV provide a test-based logo policy, which provides a thorough evaluation and assures minimum conformance, but that is not true for every middleware package.

Brazil is replacing its PAL - system M (PAL-M) with the ISDB-T's physical layer and new features [13], which resulted in SBTVD. Compliance with the latter is partially assured by an official test suite [14] and proprietary approaches [15] created for its middleware, known as Ginga or DTV Play (Ginga profile D) [9,16–19]. However, many subsystems do not have official test specifications, thus forcing manufacturers to perform such a task. Besides, there is no formal logo procedure for Ginga: conformance is evaluated with a "self-certification" strategy [20], i.e., manufacturers must assure compliance by themselves. Moreover, it usually leads to proprietary test suites that usually cover Ginga and program-specific information (PSI) and service information (SI) tables [15, 21], since physical layer [22] and reception [20] are provided by deeply-assessed devices.

Despite testing efforts for DTV receivers worldwide, one issue remains: field problems are frequently reported in any DTV system, directly affecting end-user experiences and after-sales costs. Briefly, a field problem in the test area can be defined as a malfunction event that happens outside a laboratory, usually on user premises, which often compromises the whole user experience. In the DTV area, a field problem usually prevents a user from watching TV programs by interfering with audio, video, or other data. Additionally, such events are typically handled by DTV-receiver manufacturers because their products must behave as specified and then provide low failure rates. Moreover, other questions come out: *what are their root causes? Are they related to development errors or other factors not yet identified?*

Recently, further investigation in SBTVD networks, carried out by a DTV manufacturer named TPV/Envision, revealed that several field problems were caused by incorrect information sent in broadcasters' transport streams (TSs) [23], mainly related to PSI/SI [23,24], source coding [25,26], Ginga applications [27], and synchronization [23, 28]. Indeed, that means wrong data on transport level, where configuration used for DTV receiver operation is borne. This significant discovery further clarifies the field-problem discussion. It can also be extended to other standards since the same elements are used: TSs from the moving picture experts group (MPEG), PSI/SI tables, compressed media, and interactive applications. As a result, this paper is the first to present a robustness evaluation methodology for DTV receivers, which can benefit test practitioners worldwide.

One could also argue that DTV manufacturers are not responsible for those faults, and broadcasters should maintain the correct configuration and be responsible for eventual problems. However, this is not the usual opinion of end-users and retailers because they do not understand such details and return products in case of faulty behavior. Therefore, it is more profitable and practical for manufacturers to develop devices that check data received from input signals. Moreover, amendments should be made before using them, leading to verification procedures in modules and test systems capable of simulating the mentioned situations and scenarios, thus revealing weaknesses and providing opportunities for increasing receiver robustness. Consequently, the currently available literature in system and software testing lacks research focusing on those aspects to target evaluation methodologies for DTV receivers.

Furthermore, a more detailed analysis revealed that the noticed field problems were somehow caused by random events, a combination of incorrect information, sometimes slight or non-significant deviations, and how controlling software is developed and interacts with it. Indeed, anything different from what is expected may cause severe malfunction, depending on specific conditions. In that sense, it resembles how fuzz testing is performed: random data generation as input to a computer system to find errors and security vulnerabilities [29, 30]. More specifically, a sort of guided fuzzing [31] is actually in place, given that underlying systems and protocols dictate restrictions and scenarios. Consequently, such an insight sheds light on this matter and reveals how to anticipate these kind of problems, which may be done during the development phases.

DOI: 10.1002/stvr

Usually, approaches for avoiding field problems include traditional testing batteries with functional, integration, conformance, and field tests [15, 21, 32]. In addition, they may even include some tests targeting robustness [32], but that is rarely done systematically, and we have no knowledge of a system aiming to prepare a device for real error scenarios. Moreover, the mentioned tests are carried out during development phases or after a field problem is reported. However, the latter is too late as product return, or replacement is on its way, or technical assistance has already been provided. Consequently, a structured approach aiming at robustness regarding field problems and based on the way they emerge seems necessary and presents itself as a gap in the verification area.

The scenario above inspired the present study, which proposes a paradigm shift regarding DTV receiver evaluation methodologies and extends a previously published paper [33]. Instead of checking whether it behaves as expected, we exploit state-of-the-art fuzzing techniques to verify the DTV receiver response against admittedly inaccurate or inconsistent data, generated with an approach based on guided fuzzing, to identify improvement opportunities concerning robustness. A non-compliance testing methodology was developed and implemented, allowing commercial receivers experimentation and providing verification results. The specific research challenge involved here may be stated as developing a methodology to evaluate DTV receivers under the wrong configuration and unforeseen scenarios while providing means for device enhancement. Additionally, the proposed methodology defines a complete evaluation approach, which includes a testing environment, audio and video verification algorithms, and a strategy for test creation. Regarding the latter, previous field problems, user interfaces of commercial equipment, and sensitive data are used as starting points for guiding and devising tests. However, all that can grow and evolve with time in order to provide a mature testing framework. Consequently, new verification cases are expected as the understanding of the associated environment advances so that the resulting verifier can be up to date and also continuously improved with checks that distantly resemble the initial ones.

1.1. A brief Discussion on Motivation

As hinted in the Introduction, there is a gap regarding DTV-receiver testing, which traditionally does not address misconfiguration in head-end equipment. Indeed, current testing resources only consider the correct formation of DTV signals; however, those systems have an attractive condition that can lead to incorrect configuration data: due to the intersection of different subsystems, such as audio and video source coding, multiplexing, and SI/PSI, the same equipment can be even used for different DTV transmission systems, which human operators then configure. In addition to the enormous configuration options available, this scenario can lead to DTV signals with wrong data, which is not usually expected by commercial receivers, focusing on conformance to what should be transmitted. Consequently, it is not unlikely that wrong configuration is incorrectly handled in receivers, which may lead to non-operating devices or compromised output interfaces (i.e., video and audio) as perceived by end users.

Note that such occurrences then give rise to a chain of events whose peak constitutes the replacement or even the return of a product, bringing manufacturers financial loss. Although one may claim that it is only a matter of "correcting" the faulty information, such an action is not as easy as initially assumed. Sometimes, small broadcasters can not even be contacted; if they are, they may not know what to do. This way, the onus of such occurrences is often left to manufacturers.

Consequently, we went through existing state-of-the-art techniques in the search for a framework capable of handling such scenarios. However, we soon realized that the current literature did not provide what was needed [15, 21, 30–32, 34–42]. In addition, no tool had already been developed with such a goal.

However, by analyzing real field problems and considering how they arise, why not tackle them and anticipate what will probably happen in the field? Moreover, can we create a methodology complementary to usual verification procedures, e.g., bounded model checking [43, 44] and conformance testing [45, 46], so one does not have to wait until something happens on user premises? In addition, given the way field problems arise, that seems to be the case and undoubtedly involves fuzzing techniques to mimic the randomness identified in their constitution.

In summary, the motivation of this work became evident and led to a novel verification framework that addresses a previously ignored condition: creating a scenario where signal configuration is purposefully corrupted to reveal receiver fragility, which is identified using the audio and video outputs available to end users.

1.2. Contributions

This paper makes four main contributions. First, it presents frequent field problems and their root causes, which have never been revealed to the best of the authors' knowledge. Second, a methodology for DTV-platform evaluation, providing a complete scenario with tools and infrastructure, is introduced and implemented, which can be applied to DTV systems worldwide. Besides, as extensively known, simple fuzzing usually presents low efficiency. However, based on how field problems usually manifest, a guided-fuzzing approach for test creation was devised, potentially revealing likely problems and dangerous scenarios. Finally, real experiments were performed for seven different receivers from five manufacturers, clarifying their significant weaknesses.

1.3. Paper organization

The remainder of this text is organized as follows. Section 2 reviews testing technologies for MPEG TSs. Then, Section 3 discusses real field problems, while Section 4 presents fuzz testing, along with a brief analysis of the proposed approach regarding the available fuzzing strategies, classifications, and techniques. In Section 5, a methodology for receiver robustness evaluation is proposed, as well as a strategy, based on guided-fuzzing, for error creation. Experiments for validating this methodology and providing statistics are shown in Section 6, some related work is discussed in Section 7, and, finally, conclusions are described in Section 8.

2. TESTING RESOURCES FOR MPEG TRANSPORT STREAMS

Settings for PAL, SECAM, and NTSC signals are restricted, with one video and, usually, two audio signals [2]. By contrast, DTV systems provide multiple video and audio streams, electronic program guide (EPG), and interactive applications [47] that, in the case of SBTVD [22], are developed in nested context language (NCL), with scripts in Lua, or Java [22, 24, 27, 48]. Moreover, SBTVD and ISDB-T [8, 49] allow different simultaneous modulation types. This way, a broad configuration is required, partly set by technicians at DTV head-ends.

Verifying receiver compliance and if MPEG TSs present the correct configuration are two necessary tasks that led to many studies. Regarding the former, southeast Europe has created a conformity assessment specification for DVB [50] to guide member countries through its evaluation. Concerning the latter, specialized analyzers verify TS compliance [51]. DVB has developed test specifications regarding bit rate, synchronization, and transmission parameters for MPEG TSs [52]. It has also devised a measurement and signaling channel [53] that allows the insertion of test data into a specific packet identifier (PID). In Finland, rules for SI tables [54] have been published, which help broadcasters comply with *NorDig* receivers [55]. Finally, a Ginga test suite has been developed [14], and operation guides have been released for DVB-T [56, 57], ISDB-T [58], and SBTVD [59–61]. It is worth mentioning that the Ginga compliance suite could be manually executed or even integrated into automated test equipment (ATE) [21].

In summary, there is no unified TS certification system because its composition may also change on the fly. Furthermore, although there is equipment for shared resources, such as essential TS formation [52], specific-information devices are rare. Consequently, broadcasters are responsible for verifying TS conformity using automated tools, monitoring equipment, and manual checking. Besides, there is also a lack of devices for performing specific checks for SBTVD, which then forces the adoption of generic [47] or limited tools [62] and further complicates TS certification processes.

Moreover, although field problems are significant and lead to high costs for after-sales departments of manufacturers, they have always been handled in a non-systematical way. Receiver manufacturers try to contact broadcasters, and the latter, when affected (e.g., loss of audience due to non-operating receivers), try to correct its configuration; otherwise, manufacturers create fallbacks on their own to prevent further costs.

Due to that, robust devices seem more convenient for dealing with erroneous information. In this context, a further investigation performed by the same Brazilian TV manufacturer mentioned before led to gathering and analyzing TSs recorded during failures to make its devices more robust. Therefore, our investigation revealed conditions that had not been anticipated, thus inspiring robustness tests that also evaluate non-compliant scenarios. In addition, the present work is a direct consequence of such effort. It seems to be the first to address the mentioned problem and creates a methodology for DTV receiver robustness under configuration errors. Finally, it favored the creation of verification equipment capable of being applied to real DTV signals.

3. FIELD-PROBLEMS ANALYSIS

Formally, a field problem is an issue that happens outside a laboratory and after the development phases of a specific device, that is, during its after-sale period. It means field problems are first noticed by final users or retailers when trying to operate a given product. It happens mainly because such occurrences usually compromise audio and video outputs and even menu graphics and auxiliary information (e.g., EPG), thus directly affecting the resulting user experiences.

Field problems can be classified into system-, audio-, video-, application- and data-related occurrences. Further, these same categories can be divided into incapacitating and non-incapacitating problems. The first classification level is straightforward and identifies the specific part most affected by a given problem. In contrast, the second level refers to its severity, that is, if the associated device can still be operated.

It is worth noticing that DTV field problems are usually caused by implementation errors or wrong configuration sent by broadcasters, the latter being the focus of the present work. In the following sections, some problems that happened in Brazil are presented, which already include their classification as explained above. Moreover, not all categories are represented as the problems explained here were chosen only for being relatively important, i.e., they are frequent or ultimately compromise receivers, and they are typically identified in roughly 60% of field problems routinely found, as reported by the R&D department of the previously mentioned TV manufacturer. In addition, to clarify some aspects specific to digital televisions, a brief description is first provided in Section 3.1.

3.1. Basic Aspects of the Digital televisions Systems

A digital television signal consists of programs multiplexed in a TS, as illustrated in Fig. 1. Each program includes component streams, which can be audio, video, data (e.g., subtitles, specific tables, and stream events), applications, and synchronization information that is known as program clock reference (PCR). Audio and video are sent in compressed format [63], due to bandwidth restrictions. Additionally, data are packed depending on their use [64], and interactive applications are transmitted with objects or data carousels [65]. Lastly, PCR information provides synchronization with a given transmitter, ensuring a suitable decoding rate. The latter is of paramount importance, given that it allows general and inter-media synchronization, with a 27MHz clock, and even generates frequencies for output interfaces [64].

Usually, in SBTVD, two programs are sent: a high-definition one, also known as primary, identified with *xx.1* [24], and targeted on fixed receivers; and a low-definition one, identified with *xx.31* [24] and targeted on mobile receivers [24]. Depending on specific cases, each program may contain more than one audio or video stream. Usually, only one video stream is transmitted. However, there may be one, two, or even more additional audio streams, which can bear, for instance, an original program's audio, in its original language, and audio description, for visually impaired people, which is composed of its original sound and scene description [66]. Nonetheless, to notify receivers about primary streams, the field *component_tag*, in the stream identifier descriptor [67], is used. For instance, in SBTVD, a primary audio stream presents *component_tag* equal to

DOI: 10.1002/stvr

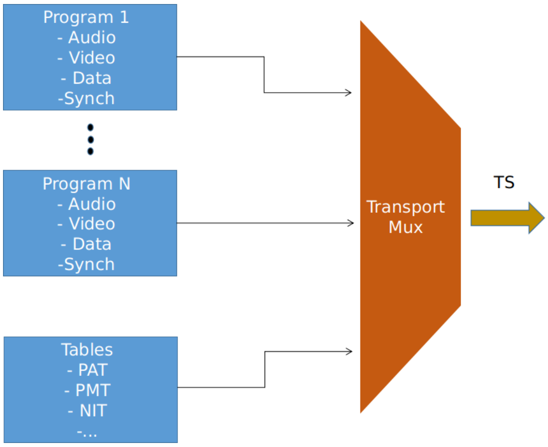


Figure 1. Basic structure of a digital television signal.

$0x10$, while the others follow the sequence $0x11$, $0x12$ and so forth [61]. Finally, other types of streams may be present, such as subtitles, object carousels, and even Internet protocol (IP) packets [64, 68].

3.2. *Video-Related and Incapacitating Problems*

This section presents field problems that primarily compromise the video output. This means the video subsystem processes the erroneous elements, leading to a condition where the affected device becomes non-operating.

3.2.1. Wrong Length of Supplemental Enhanced Information In TV services, the essential data are video, audio, and subtitles/closed caption (CC). On the one hand, the first two are usually sent as independent packetized elementary streams (PESs) [23]. On the other hand, subtitles/CC may also be borne in MPEG version 2 (MPEG-2) video [6] and H.264 [69, 70] streams, as happens in ATSC. This information is vital because many DTV platforms are employed worldwide, with differences only regarding air interfaces, which leads to the use of the same development platforms.

In one specific field problem, receivers malfunctioned when tuned to a broadcaster. After a first decoding attempt, it involved a lack of response to remote control, graphical user interface (GUI) slowness, and video absence.

A deep analysis of the related TS revealed that supplemental enhancement information (SEI) packets of type *user_data_registered_itu_t_t35* [69] presented wrong size in headers of H.264 network abstraction layer (NAL) units [69]. In SBTVD, such messages are used for active format description (AFD) [25], but the faulty stream bore CC content [70, 71], with a divergence of 3 bytes between message length and CC-data size. Consequently, parsing procedures were unable to identify subsequent NAL units. This problem occurred due to faulty professional equipment, which resulted in a fix request to the respective manufacturer and CC-feature disabling. Moreover, it could have been avoided if the actual data-size (*cc_count*) [71] had also been taken into account.

One may notice that this problem could have happened in other DTV systems, such as ATSC, DVB-T, and ISDB-T, because they also adopt H.264 and use commercial platforms. Moreover, even if transmitted in ATSC networks, such length information would still be wrong. Finally, the DTV markets in the United States and Europe also employ the platform that presented this behavior.

3.3. *System-Related and Non-Incapacitating Problems*

This section presents field problems that primarily compromise the whole DTV system in receivers, affecting all subsystems. However, the associated devices can still be operated.

DOI: 10.1002/stvr

3.3.1. Clock-Reference Values Smaller Than What Was Expected This problem is very aggressive and can affect many DTV subsystems in a commercial receiver, thus deeply compromising its operation.

A broadcaster sent a primary service that caused halting and slow video, making it impossible for end-users to watch the respective channel. A brief analysis of the referred TS revealed that some PCR values [23, 64] were odd. In particular, there were two PCR lines: one adequate regarding the available presentation time stamps (PTSs) [23, 64], and another one presenting values lower than expected. Besides, there was no PCR discontinuity [23].

From a receiver's point of view, PCR values must increase and, ideally, match a local 27 MHz clock. However, when sudden smaller values appear, traditional algorithms identify that the current clock is too fast. The associated phase-locked loop (PLL) [64] then tries to get synchronized, reducing its current frequency. Consequently, a generalized malfunction occurs, affecting video and audio reproduction, stream synchronization, and output interfaces [2, 64].

Further investigation revealed that a new multiplexer was being tested, including PCR information into the same packets used by the original equipment. In addition, in another instance of the same problem, the same multiplexer generated two different time bases and added them to packets with the same PID. It can be seen in Fig. 2, where one may notice two distinct PCR lines: upper and bottom parts of that figure, with red squares (other colors mean data related to PTS and decoding time stamp - DTS). The y axis shows PCR values, in milliseconds, with the two mentioned distinct lines, and the x one shows PCR packets along time, also in milliseconds. If the upper line were the only one, the internal PLL in receivers would lock on it with a frequency that matched PCR values from left to right (e.g., from 100ms to 200ms and so on). However, the lower line provides sudden changes in its frequency (e.g., from 100ms to 50ms and then back to 200ms), which tampers with the underlying synchronization algorithm.

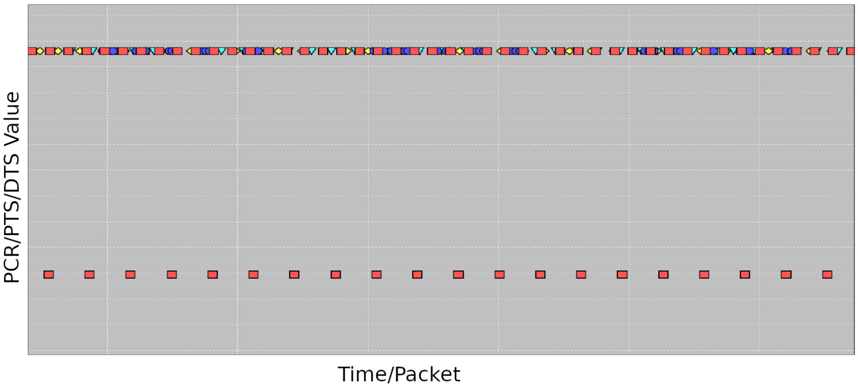


Figure 2. Field problem caused by incorrect PCR values: red squares mean PCR data, while the others are presentation and decoding time stamps.

The problem explained above is tough because PCR is used for synchronizing local 27 MHz clocks [23]; however, wrong values could be ignored with filtering [72]. Finally, PCR is present in any standard based on MPEG TSs, including ATSC, DVB-T, and ISDB-T.

3.3.2. Time Gaps Larger Than What is Recommended Between Two Table sections Usually, DTV standards define transmission rates for PSI/SI tables [24, 60, 61], such as program association table (PAT) and network information table (NIT), in order to make related information available according to its relevance. PAT sections, for instance, are usually advised to be transmitted in intervals of 100ms, while time offset table (TOT) ones may be delivered every 5s.

During a particular glitch, television sets suddenly suffered a channel change when tuned to a specific broadcaster. Indeed, if its physical and virtual channels [24] were YY and XX, respectively, then DTV devices presented changes from $XX.1$ to $YY.1$ and back.

DOI: 10.1002/stvr

When reproduced in the laboratory with the faulty TS, an issue related to a non-conformity regarding NIT was discovered. Although virtual channel information in TS descriptors was sent as indicated by standard ABNT NBR 15603 [24], NIT sections were sometimes transmitted according to a time interval slightly higher than 10 seconds, which is the limit imposed by the same standard. Such a configuration caused filter timeouts in receivers, and, in the absence of that table, the respective physical channel was used as a fallback procedure. So, if a timeout happened, the physical channel was used; otherwise, the virtual channel was displayed in an alternating pattern. Given that such a handler incurred a channel modification, end-users assumed that a real channel change was in course.

One may notice that changing a channel was a specific feature of the identified DTV receiver model; however, ABNT NBR 15603 [24] indicated a mandatory transmitting period no longer than 10s. Besides, NBR 15608 later suggested the transmission of NIT sections every second, which is usually done by several broadcasters [59, 61]. Finally, this is an example where developers strictly follow what is indicated by a standard, without error margins, a more elegant approach, or a more sophisticated fallback procedure.

In summary, this problem was a combination of wrong configuration and fragile software implementation, which is very common in DTV environments. Finally, the table transmission rate is a concern in every DTV system, and platforms must provide a way of handling wrong setups.

3.4. Data-Related and Incapacitating Problems

This section presents field problems that primarily compromise the auxiliary data sent in a TS. The erroneous elements are processed by the SI/PSI subsystem and lead to a condition where the affected device becomes non-operating.

3.4.1. Conditional Access Information Transmitted in a Free-to-Air DTV channel Conditional access information is usually employed in cable and satellite TV systems to restrict access to entitled subscribers. In summary, such a piece of information is split into a specific table, called conditional access table (CAT), related descriptors, entitlement management messages (EMMs), and entitlement control messages (ECMs). However, due to their characteristics, those are not usually employed in free-to-air and terrestrial television systems.

In this particular problem, from time to time, TV sets did not display the primary service of a broadcaster. When that happened, they crashed, and a complete system reset was required to recover regular operation. An investigation conducted on a TS presenting this problem revealed conditional access information in CAT sections, conditional access descriptors carried by program association table (PMT) sections, and fields *conditional_access_mode* in service description table (SDT) sections [24], even though TS packets with the PID associated to EMMs available in CAT sections were not present. Consequently, DTV receivers tried to initialize their decryption circuits without real encrypted data, which eventually caused the mentioned behavior.

Indeed, information related to conditional access should be ignored in free-to-air terrestrial transmissions, at least in Brazil and many other countries. However, DTV platforms are used in many conditions and markets (horizontal and vertical ones), leading to code for that scenario being run in non-compliant conditions, which may also happen in other DTV standards.

3.4.2. Non-existent Services Specified in NIT DTV systems deeply rely on PSI/SI tables; however, not all tables must be transmitted, but a subset of them usually includes PAT, PMT, NIT, SDT, and event information table (EIT) sections. In particular, NIT sections bear essential information regarding services and their classification, which usually drives structure creation in a receiver's memory.

During a specific problem, which repeats itself once in a while, receivers usually have their databases corrupted and consequently stop working when they are tuned to a given channel or do not display services. Following the acquisition of the respective TS, it was noticed that NIT sections carried a service list descriptor with several "ghost" services, which were not present in SDT and PAT sections [24].

DOI: 10.1002/stvr

After contacting technicians, the latter informed us that many tests with different DTV services had been performed, and the related information was never removed. Although SBTVD and other DTV standards (e.g., DVB-T and ISDB-T) report that PSI/SI tables must be consistent, receivers may perform cross-checking to use only data that seems correct.

As a final remark regarding DTV systems, much information is often shared among different tables, which must be consistent and checked before transmission. Moreover, information that presents many configuration possibilities is highly prone to error and should be carefully cross-checked.

3.5. Data-Related and Non-Incapacitating Problems

This section presents field problems that primarily compromise auxiliary data sent in a TS. The SI/PSI subsystem processes the erroneous elements, but the affected device can still be operated.

3.5.1. Inconsistent Encoding of Audio and Video Streams In DTV, there are two layers of media information: multiplex, in PSI/SI tables and usually employed for decoder configuration, and media, in elementary streams (ESs) and accessed only by decoders. A disagreement between those is a classic problem that may lead to audio and video situations.

Many transmitters used in Brazil have been configured with information used in Japan, where the audio data transport stream (ADTS) container [73] was initially adopted. Consequently, low-overhead MPEG-4 audio transport multiplex (LATM) [74] streams are sometimes transmitted with ADTS specified in PMT sections (field *stream_type*), which may be wrongly performed by technicians [75]. Then, decoders may be erroneously configured, or audio may not even be decoded due to a lack of support to ADTS or algorithm allocation inconsistent with LATM.

Indeed, as technicians can provide information regarding elementary-stream format, other DTV systems may also suffer from that. In contrast, decoding is possible if an implementation is used as the required information is available on ESs. Finally, a receiver could perform decoding attempts with all available formats as a last resource. In the past, a similar problem occurred in DVB-T receivers: MPEG version 4 (MPEG-4) part 10 video being tagged as MPEG-2.

3.5.2. Audio PID Specified in PMT Sections but not Present in a Given TS In summary, TSs are collections of packets with different PIDs, which are multiplexed by time division [64]. Nonetheless, no central element exists that provides information regarding all of them. However, instead, that is split across PSI/SI tables, while other present fixed PIDs, such as PAT, with PID equal to $0x00$, and NIT, with PID equal to $0x10$ [23, 24].

In this field problem, which is very common and happens almost every year, DTV receivers were not presenting audio when tuned to a specific broadcaster, even with four audio streams being transmitted. Indeed, the audio stream with *component_tag* equal to 16, which should be the primary one [20], was not present, i.e., TS packets tagged with its respective PID were not transmitted. Besides, crash events may even be perceived, depending on the assigned PID and its absence or different content.

PIDs announced in PMT sections can be configured in different ways, which must be consistent with what is currently provided by a multiplexer. If PIDs in PMT sections change for some reason, that information will not be referenced in a TS, i.e., those packets will be present, but receivers will be unable to recognize them. Again, that is a common problem in almost all DTV systems worldwide, which are based on MPEG-2 TSs.

When an audio or even a video stream can not be decoded or nothing comes out of its PID filter, one possible action could be to try a stream with the next *component_tag* or, if that information is not sent in the stream identifier descriptor [67], the next stream specified in a given PMT section, in a subsequent table loop.

3.5.3. Closed-Caption Stream with Wrong Component Descriptor PSI/SI tables present fixed and dynamic information, with descriptors and table fields, respectively [23, 67]. The former is used for special or temporary cases without fixed information and tools for quick and transparent standard updates.

DOI: 10.1002/stvr

In Brazil, often, TV sets do not present CC when tuned to some channels. A brief analysis showed that component descriptors [24] were sent in PMT sections; however, the fields *stream_content* and *stream_type* were set with values reserved for future use, according to the DVB's standards [67].

There is an intersection between DTV systems, as most use MPEG TSs and PSI/SI tables; however, each has specific definitions. Consequently, many devices usually come with default configurations for a different system. Besides, many technicians do not analyze the SBTVD's standards and only use default settings. Moreover, fields with many options may easily lead to the wrong attribution, even with automated tools, due to implementation errors or misinterpretation.

Reserved values should be selectively used or rejected in robust DTV receivers to avoid functionality impairment. Finally, this same problem could also happen in ISDB-T networks.

3.5.4. Service Description Table with Incorrect Section Number PSI/SI tables can be very long and are divided into sections of at most 1 or 4KB [22,67]. Two table fields report their organization: *section_number*, with the current section, and *last_section_number*, with the last set section.

When tuning to digital channels, DTV sets store either the high-definition or the low-definition [20] program, whose choice depends on the moment of a scan. As a result, several televisions end up tuning to the mobile signal and displaying low-resolution video. Moreover, most DTV sets block mobile signals due to their low resolution. This issue leads receivers to ignore the mobile-signal blocking, which exemplifies the extent of a given field problem.

A deep analysis revealed that the associated SDT [24] specified two services in two different sections (*section_number = 0/1*), but informed only one section (*last_section_number = 0*). Some devices would then show content from the first received section and ignore the other. Indeed, this problem happens due to a combination of receiver fragility and faulty human operation. The latter is because one decided to split service descriptions into different sections, which may happen to other tables.

One way to increase receiver robustness would be to avoid using the field *last_section_number* [24] directly and increase the priority of *section_number*: the final table would always be the union of all sections, no matter the last one. Finally, such a problem can affect any DTV system based on MPEG TSs, due to the extensive use of SDT.

3.5.5. Different Frequencies using the Same Virtual Channel Virtual channels may cause problems as they access DTV services, and SBTVD standards do not clearly explain this part. DTV broadcasters from the very high frequency (VHF) bands should send virtual channels in their ultra-high frequency (UHF) transmissions, with the same VHF number [20]. In contrast, new stations should send virtual channels set with their physical ones. Different operators sometimes set the same virtual channel numbers, which makes them absent while browsing DTV programming in receivers. Indeed, the preferred index key for navigation is a physical channel number, which is unique. However, that is implementation-related and should be encoded in software.

Finally, this error is fascinating because it goes beyond a single DTV operator and involves an entire network. Although transmission monitoring is performed in Brazil, it is an arduous, highly open, and dynamic task.

3.6. Application-related and non-incapacitating Problems

This section presents field problems that primarily compromise the application subsystem. Consequently, a middleware module processes the erroneous elements, and the affected device can still be operated.

3.6.1. Compressed Ginga Application With Incorrect Descriptors One essential aspect of DTV systems regards distributing and running interactive applications [9, 10], usually sent in object or data carousels [10], compressed or not. The latter is also informed in PSI/SI tables and an object carousel.

Sometimes, DTV sets identify a Ginga application [27] in a TS, but its loading procedure never completes. As a result, it is never executed, although some televisions keep announcing its presence.

DOI: 10.1002/stvr

After analyzing object carousels, it was found that applications were in compressed format, even though that was not shown in the related carousel identifier [65] and compression type [76] descriptors. As a result, some receivers attempted to run them in compressed format.

Although compression was not informed, some devices did run those applications. When its entry point was not found, they tried to decompress it, as a fallback procedure. Further analysis revealed that there was a *compressed module descriptor* [65], which is specific to DVB, but not the *compression type* one [76], as used for ISDB-T and SBTVD. Again, it could also affect other DTV systems, such as ISDB-T.

3.7. Discussion

All field problems presented in this paper are real and were identified and characterized during terrestrial transmissions in Brazil. Simultaneously, many of them were also noticed in networks and receivers conforming to other standards. Besides, all analysis results reported here were confirmed and led to changes in transmission-equipment configuration and receiver software. Moreover, some also resulted in bug fixing related to head-end equipment's source code.

Those non-compliance events indicate errors caused by malpractice in configuration, impairing fragile DTV receivers. Nonetheless, there were also problems caused by faulty equipment, which is not usually expected, and incorrectly- or carelessly-set table fields, which present many options and versions.

Many DTV subsystems may present wrong information, mainly regarding ESs; synchronization and timestamps; PSI/SI tables, including descriptors, the relation among streams, and multiplex configuration; interactive applications; and interrelation among fields. In addition, their emergence presents some randomness, which arises from unforeseen interactions among configurations available for manual and even automatic setting, receiver-code development, transmission equipment development, standard restrictions, and configuration-data structuring. It became apparent with the analysis performed due to the field problems shown above, which also shed some light regarding ways of predicting them and providing handling algorithms. Finally, such problems usually lead to noticeable symptoms: the absence of audio or video; video freezing, where images do not change; video flickering, where images are suddenly black or white; frame skipping, where frames are lost; artifacts; and audio discontinuities. Consequently, device robustness improves after-sales cost reduction if fragility is probed, monitored, and then handled.

Concerning DTV-system robustness, one may argue that anything may happen, following their characteristic randomness. However, it is possible to take real issues and create peripheral or similar non-compliance random events, as related data are also susceptible to error in a guided fashion. Besides, elements in GUI and highly-flexible data are essential and prone to problem emergence. Therefore, to evaluate and improve DTV receivers' robustness, it is necessary to create non-compliant TSs based on the mentioned ones and already incorporate the randomness characteristic of this kind of field problem, which directly leads to fuzzing techniques (see the next section).

Another point regards end-users: do they care about the problems presented here? The answer is: yes, they do. In addition, the explanation is simple. In most problems presented here, an end user could not watch TV due to the problematic video or audio presentation caused by them, which led to massive access to TPV/Envision's after-sales department and product returns both by retailers and end-users. Moreover, when a DTV receiver is not compromised, the impossibility of using affected features leads to a low-quality perception. It was also noticed during the data collection performed in the context of this study.

The previous statements inspired the methodology presented here, which will be introduced and developed. Moreover, a given standard restricts conformity assessment; however, non-conformity takes a vast range of possibilities, even if bounded as suggested: previous problems, user-interface elements, and sensitive data, which leads to some automation. Besides, it should be device-independent due to lower costs and higher applicability.

Finally, most problems reported here are not restricted to SBTVD and may happen in other systems, such as ISDB-T, ATSC, and DVB-T. Consequently, a methodology able to assess receiver fragility would present broad applicability.

4. BACKGROUND THEORY: FUZZING AND RELATION WITH THE PROPOSED APPROACH

Fuzzing is a popular testing technique, developed in the 1990s, that relies on the random generation of data for evaluating computer systems [30]. Its simplest version requires little knowledge of target systems, even without the related source code, and can be easily scaled to large arrangements. It can be executed in four steps: test case generation, i.e., random input data generation; test case execution; state monitoring; and exception analysis. However, such an original simplicity also presents an important disadvantage: low effectiveness regarding bug identification. Consequently, it has also evolved a lot since its creation, with techniques that include guided testing, mutation, program analysis, machine learning, slicing, and symbolic execution [31].

Test quality, in fuzzing, is of paramount importance, including input format and granularity, which should, with high probability, lead to a program failure. Fuzzing may rely, as inputs, on files, communication data, or even executable binaries; however, in the case of DTV systems, DTV signals, according to a given protocol, take place, which usually converge on the TS level [23]. Consequently, fuzzing in DTV networks inevitably involves the generation of broken TSs so that failures in specific subsystems are accessed. Besides, when a test case is run, fuzzers need to monitor program states or outputs to identify exceptions and crashes, where even specific tools for program monitoring can be used [31]. Nevertheless, as the present works focus on DTV receivers, every symptom converges to their standard outputs: audio and video.

Fuzzers can also be classified in different ways, according to the aspect in focus. Regarding input creation, Fuzzers can be seen as either generation- or mutant-based elements [36]. The former involves transformations over a reference input, while the latter employs system specification. Although simpler, mutant-based fuzzers usually lack coverage and are often less effective, with inputs that largely deviate from what is expected. Nonetheless, as DTV systems already provide deep specifications and reference software, a generation-based fuzzer has everything expected to create test cases and seems feasible. When code dependence comes into play, a fuzzer can be classified as white-, grey-, or black-box. White-box Fuzzers have access to source code, while black-box ones do not. In addition, grey-box fuzzers usually employ program analyzers to obtain knowledge of a program. That being said, when evaluating DTV receivers, their source code will probably not be available. However, there are open standards, and the way receivers are implemented and their resulting processing chain are well known. Consequently, any analysis can be classified at least as a grey box. Next, program exploration can also be taken into account. Direct fuzzers result in test cases that aim at specific code and paths, i.e., known vulnerable parts, while coverage-based ones try to increase code coverage. In the present case, we envision a test targeting the DTV processing chain and, more specifically, fragile parts involved in field problems. Consequently, we look for a coverage-based method [37]. Finally, there are dumb and smart fuzzers. The former usually only rely on data range and blindly generate test cases. At the same time, the latter holds a more profound understanding regarding what is being handled, which allows going further in code paths and processing chains. Indeed, one should notice that we can start from knowledge regarding standards and field problems, the latter tending to repeat themselves or occur in similar areas, and the most fragile parts (see Section 3), which can be encoded into our fuzzer, then making it a smart one.

In addition, higher code coverage is also desirable, at least over time and based on new knowledge. Consequently, there must also be a manner of increasing tested parts and modules based on some fragility criteria. This way, we both improve efficacy and coverage. Besides, by taking into account what was presented above and having in mind our target application, i.e., DTV receivers, we can summarize the main questions involved in fuzzing and their respective answers in the light of the present problem:

- inputs are obtained based on the knowledge of DTV standards, known field problems, and fragile spots, which directly leads to improved efficiency when compared with purely-random strategies;

- test cases are generated based on the DTV processing chain and known implementation strategies, targeting potentially-vulnerable code parts;

- DTV receivers can be tested by sending signals (TSs) and monitoring their audio and video outputs, with test time and parameters adapted to specific DTV standards.

Moreover, a smart fuzzer designed for DTV purposes can be depicted as shown in Fig. 3. A generator, fed with information for smart operation, creates input TSs and presents them to a device under test. Then, an output monitor, focused on audio and video outputs, judges if a violation is present, resulting in other tests or error registration (bug). Besides, the smart information in Fig. 3 also includes data for somehow creating malformed structures, in terms of organization, or even hit corner or cryptic cases in DTV specifications, which leads to a grammar-based approach [34,38,77]. The latter both focus on specific structures and provide apparently-valid data for the DTV processing chain. For example, the structure of PMT sections is shown in Table I [23], which is, undoubtedly, a possible target for the proposed grammar-based guided-fuzzing approach. Furthermore, as can be inferred from the problem explained in Section 3.5.1, the field *stream_type*, which is 8-bits wide, should undoubtedly be fuzzed, as a myriad of problems may be introduced, such as disagreement between content and signaled encoding and packet absence. Finally, many other essential fields can also be fuzzed, such as *descriptor* and *last_section_number*/*section_number*.

Syntax	**Bitwidth**
TS_program_map_section() {	
table_id	8
section_syntax_indicator	1
‘0‘	1
reserved	2
section_length	12
program_number	16
reserved	2
version_number	5
current_next_indicator	1
section_number	8
last_section_number	8
reserved	3
PCR_PID	13
reserved	4
program_info_length	12
for (i = 0; i < N; i++) {	
descriptor()	
}	
for (i = 0; i < N1; i++) {	
stream_type	8
reserved	3
elementary_PID	13
reserved	4
ES_info_length	12
for (i = 0; i < N2; i++) {	
descriptor()	
}	
}	
CRC_32	32
}	

Table I. Structure of PMT sections.

It is worth noticing that a DTV standard is indeed a sort of network protocol, as the global system for mobile communications (GSM) [77]. From the receivers' point of view, it is summarized on the structures and information available in TSs, which are then used for the entire DTV processing chain. However, this chain does not include protocol responses on an associated channel as it is a simplex system (there are no responses) [22]. Indeed, results are only visible to internal receivers' structures or via their outputs. So, incorrect data can lead to lasting and deep effects that may even echo across different modules, with the potential for compromising the resulting user experience. However, ultimately, those errors can be captured via audio and video interfaces.

As examples of protocol fuzzers, we can cite AutoFuzz [42] and AFLNet [40]. The former can learn a protocol and modify communication sessions, while the latter also fuzzes real messages with a learning algorithm. Indeed, protocol fuzzers often present a target area [39, 41], but they are not usually focused on specific standards. In addition, they rely on protocol conversations (message exchange) between a client and a server, which is not the case with DTV systems. In contrast, the present approach focuses on the latter. It provides a structure compensating for its simplex construction with real problem scenarios, fragile processing parts regarding the associated data, and head-end configuration aspects. When response messages [40] are unavailable, the associated communication structure and data relation can be better tackled with a tuned approach.

Although the course of action taken here may initially seem a restriction, there is a myriad of different DTV systems around the world that can benefit from its use, which is a significant result: ISDB-T, ATSC, DVB-T, digital terrestrial multimedia broadcast (DTMB) [78], digital multimedia broadcasting (DMB) [79], and SBTVD, at least, as many building blocks across them are the same. In addition, the proposed methodology may also be applied to satellite and cable networks, such as digital video broadcasting satellite (DVB-S) and digital video broadcasting cable (DVB-C) [3]. It can also be applied to their second-generation counterparts (DVB-S2 and DVB-C2) [80, 81], as they also employ TSs as the transport layer and depend on configuration sent by head-end equipment. Moreover, other systems with similar aspects may also benefit from such a methodology and then have a verification approach with the same essence and basic structures adapted to their contexts. Examples of that are digital radio (DR) systems, such as digital audio broadcasting (DAB) and digital radio mondiale (DRM) [82]. This way, as long as signal broadcasting with transmission configuration performed at head-end equipment (e.g., by technicians) that is decoded by receivers in user premises takes place, regardless of the underlying transport layer, the basic methodology employed here can be used.

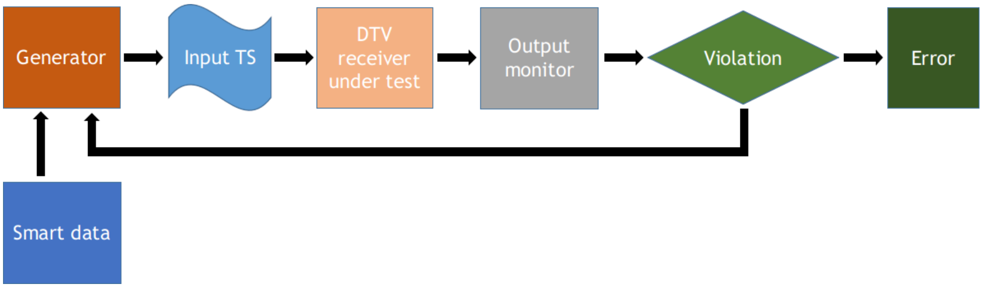


Figure 3. Block diagram for a DTV-oriented smart fuzzer.

The following section will introduce the proposed method, based on what was presented, which includes an overall scheme that goes far beyond fuzzing-based test-case generation. It will also indicate the part that includes fuzzing to explain the link with the characteristics raised here correctly.

5. THE PROPOSED METHODOLOGY FOR NON-COMPLIANCE TEST EXECUTION BASED ON FUZZING

Section 3 describes the wrong-configuration problem, which is often disregarded and may cause completely unexpected behaviors in DTV receivers. This problem happens since there is an understanding that broadcasters ensure correct DTV signals. However, mistakes do occur and may cause substantial losses. Moreover, the resulting considerations provided the basis for a robustness

DOI: 10.1002/stvr

testing methodology to create non-compliant TSs for revealing fragile code and handling strategies in DTV receivers, as follows:

- receivers often operate under non-compliant conditions, which appear in a somewhat random fashion, leading to fuzzing-based tests that submit them to those scenarios;
- there must be restrictions regarding issues and affected systems to provide a feasible implementation, which leads to a guided approach;
- an automated scheme is preferred, given the huge amount of test cases that may be created;
- a methodology should not be restricted to proprietary interfaces in order to be widely used and improved;
- a procedure should monitor what a user has access to, which leads to audio and video interfaces, in order to avoid errors that are naturally concealed or irrelevant;
- the proposed approach should focus on audio and video absence, video freezing, video flickering, artifacts, frame skipping, and audio discontinuity;
- the mentioned issues can be detected and are also the main results of almost all problems identified thus far (cf. Section 3).

5.1. General Structure Targeting DTV-Receiver Testing

This paper proposes a novel robustness evaluation methodology for DTV receivers, with a test-generation core that performs grammar-based guided fuzzing to increase device robustness without compromising user experience. Moreover, in order to provide evaluation for such elements, it also relies on a structured test environment, which is shown in Fig. 4. According to a given DTV physical layer, a Personal Computer (PC), equipped with a controlling-software module, stores and loads MPEG TSs for transmission to devices under test. Besides, its first instance could be the SBTVD's one (i.e., ISDB-T) [8, 22]. Indeed, it is essential to use a proper interface, so its reception chain is used and problems due to interrelation among different modules are then stimulated and identified. Meanwhile, the same PC software configures a DTV receiver under test through an infrared interface, which then tunes to a service that provides configuration. This way, problems in a subsystem not strictly related to decoding, reception, or configuration but also caused by them may be tackled, such as channel navigation, EPG construction, and CC activation, due to an execution that follows what happens in real operation.

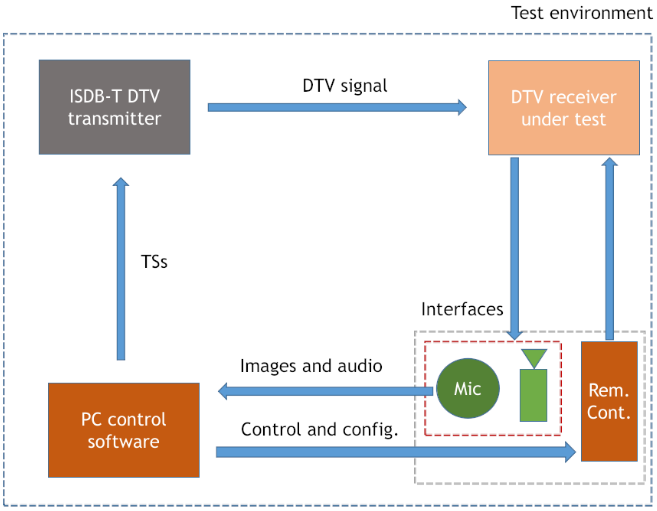


Figure 4. Diagram of the test environment employed by the proposed evaluation methodology.

In order to provide stable TSs capable of being correctly transmitted, received, and analyzed, they must have a minimum duration to accommodate tuning, pre-rolling, and signal stabilization, which is based on empirical evidence. Indeed, some transitory period is noticed after tuning, which may even take 2 ~ 3s. While signals are decoded, video and audio are captured by a camera and a microphone. Besides, this camera may not be high-quality because the main goal is to identify easily-noticeable problems and general malfunction. The most crucial aspects are correct time resolution for data acquisition and relevant reference data to favor problem identification.

It is essential to mention that one can still argue that audio and video could be directly captured. In this regard, we have many situations, but we will discuss just some of them. First, many TVs do not present video outputs as they are terminal points and are mainly used for viewing. Although some present outputs for monitoring purposes, that can not be taken for granted. Moreover, signals output from those interfaces suffers additional encoding, e.g., composite broadcast video signal (CVBS) and YC_BC_R [2]. Audio outputs, in turn, are frequently present due to a simple need: audio enhancement via external sound systems. Even so, that is not always the case for low-end devices. In addition, if some direct interface with processing boards is used, such an approach would compromise device assessment, mainly when that kind of access is unavailable for non-vendors. Consequently, it is interesting to capture signals from the interfaces surely available to users, i.e., screen and speakers, which makes it possible to apply the developed tool to every existing device.

Next, captured media is sent to the same PC software (see Fig. 4), which uses image and audio processing algorithms to detect incorrect behavior. In this regard, such algorithms should be continuously executed during an entire testing procedure, given that errors may happen at any moment. Finally, when a problem is identified (e.g., video freezing or audio absence), its presence is indicated, stored, visually informed, and then added to a *log* file, which can later be evaluated and also included in a test report.

Audio and video assessment is a challenging task that is highly dependent on content, which may be favored by predetermined signals where differences and deviations are promptly noticed. The chosen video sequence should favor identifying the mentioned problems, including video freezing, video flickering, artifacts, and frame skipping. Moreover, a suitable audio sequence could also quickly reveal absence and discontinuity.

In order to define suitable audio and video sequences, some analysis was carried out. SBTVD defines, for fixed reception, that video must be encoded in H.264, restricted to profile *high* and level 4.0 [25, 69] and with maximum resolution of 1920x1080, at 29.97 Hz, or 1280x720, at 59.94 Hz [25]. Besides, an empirical evaluation revealed that video freezing might involve as few as two consecutive frames with the same image, while flickering may be restricted to only one incorrect black or white frame between two correct ones. Artifacts may affect only a small part of a frame or include many consecutive ones, and frame skipping may involve the loss of many frames, usually less than 1s apart, which depends on a root problem.

The chosen video sequence should provide a noticeable difference between two consecutive frames, defined as a normalized cross-correlation coefficient [72, 83] lower than 0.9, which can be recognized by regular image comparison techniques [83], in frames 1/59.94s apart in time. Besides, such a difference should ideally not present a correlation lower than 0.6, which could compromise the identification of frame skipping or problems with long occurrence periods. As a result, a notification should be triggered when the correlation between two adjacent frames is higher than 0.9. The specific computation for the correlation metric is given by

$$R_i = \frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Im_i(m,n) \times Im_{i-1}(m,n)}{\sqrt{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Im_i(m,n)^2 \times \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Im_{i-1}(m,n)^2}}, \qquad (1)$$

where the i-*th* cross-correlation coefficient R_i is computed through a current captured frame Im_i and a previous one Im_{i-1}, both with dimensions $M \times N$.

The predefined video should have a dynamic structure that provides the expected correlation, with a repetition period higher than 2s. High correlation must also be avoided, which could mask a problem. In fact, such findings (repetition period and correlation) arose empirically, based on video freezing and flickering, artifacts, and frame skipping events noticed in the problems in Section 3. After extensive testing, a compound moving structure with a red square and black circles rotating

DOI: 10.1002/stvr

around a screen's center was chosen, as illustrated in Fig. 5. The red square and black circles rotate around a small central circle, with a period of 2.1s. If no video exists, correlation is high (> 0.95), while freezing also results in a high correlation among frames. Artifacts and flickering cause a noticeable difference between 0.7 and 0.5, and, finally, frame skipping usually results in a very low correlation (< 0.5). Moreover, a camera capturing the output video should also be fast enough to allow problem recognition.

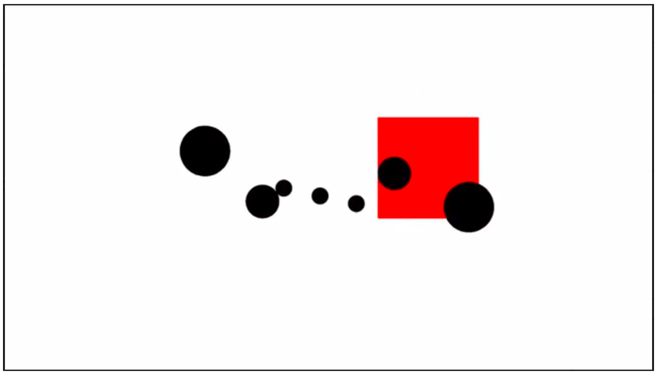


Figure 5. Reference video for automated analysis.

The adopted audio strategy, in turn, is based on transmitting stereo audio with a different frequency tone for each channel: 1 KHz for the right and 5 KHz for the left channel, which provides selective identification of audio absence or discontinuity. Indeed, audio symptoms in the analyzed field problems (see Section 3) were restricted to those two possibilities.

It is worth noticing that the chosen audio and video sequences will undoubtedly result in fast algorithms. Otherwise, for instance, if a video verification procedure took more than 1/59.94s, a test sequence might finish while the analysis was still running, which could also be true for audio procedures. Besides, extended analysis periods would also result in more memory requirements. At least another memory chunk of the same size would be necessary due to analysis-time overlap: one for the current and another for the next analysis procedure. That being so, memory needs would be twice the amount initially predicted. A simplified representation of the proposed methodology is shown in Algorithm 1, where *eval_video_and_log_error*() includes evaluation procedures for video freezing and flickering, artifacts, and frame skipping, while *eval_audio_and_log_error*() includes evaluation procedures for audio absence and discontinuity.

5.2. Test Generation Based on Fuzzing

The overall structure of the proposed verification methodology was shown in Section 5.1, but there is a vital aspect still left aside: test creation. That being said, if a nonconformity is anything apart from what is specified and randomly appears, how can one systematically create tests? Indeed, this inquiry already gives a clue of how that can be done: fuzzing, which generates random inputs based on grammars for reaching specific code.

In summary, we have employed grammar-based fuzzing, with a vast set of rules for specific parts of interest of a transport stream. However, it is worth noticing that there is no grammar for entirely constructing a transport stream but stead for fields of interest, which are fed to our fuzzing engine. For example, let us analyze the grammar for the field *program_number* in PMT sections (see Table II). It is included in the fuzzing model developed for our study and can be described, in extended

Algorithm 1 Description of the proposed methodology.

```
Require: TSs $\{ts_1, ..., ts_N\}$, TS length $LEN$, and $log\_file$
  $ts\_time \leftarrow LEN$
  $delay \leftarrow 0.1 \times ts\_time$
  $ts\_time \leftarrow ts\_time - 2.0 \times delay$            ▹ useful period
  for all $G \in \{ts_1, ..., ts_N\}$ do
    $load(G)$                                 ▹ read TS from file
    $configure\_device()$                     ▹IR commands
    $transmit\_ts(G)$
    $wait(delay)$                             ▹wait for a stable part
    $init\_time \leftarrow get\_time()$       ▹TS initial time
    $frame\_vec \leftarrow \varnothing$
    while $(get\_time() - init\_time) < ts\_time$ do
      $synch\_frame()$                        ▹one image per frame
      $frame\_vec \leftarrow frame\_vec \cup get\_video\_frame()$
      $eval\_video\_and\_log\_error(frame\_vec, log\_file)$
      $sort\_frames(frame\_vec)$              ▹remove the oldest one
      $curr\_epoch \leftarrow get\_audio\_epoch()$
      $eval\_audio\_and\_log\_error(curr\_epoch, log\_file)$
    end while
  end for
  return $log\_file$
```

Backus-Naur form [84], as below.

$$
\begin{aligned}
program\_number \;&=\; \text{``}original\_network\_id\text{''}, \\
&\qquad service\_type, \\
&\qquad service\_number\;; \\
service\_type \;&=\; \text{``01''} \mid \text{``10''} \mid \text{``11''}\;; \\
service\_number \;&=\; \text{``001''} \mid \text{``010''} \mid \text{``011''} \mid \text{``100''} \\
&\qquad \mid \text{``101''} \mid \text{``110''} \mid \text{``111''}\;;
\end{aligned}
$$

In this context, "*original_network_id*" is obtained from other tables and is a terminal symbol. At the same time, *service_type* and *service_number* are non-terminals constructed with information that is not adequate for TV services or exactly ordered as specified by the underlying DTV standards. Another example is the component descriptor, mentioned in the problem field reported in Section 3.5.3 and has the grammar shown below.

$$
\begin{aligned}
component\_descriptor \;&=\; \text{``01010000''}, \\
&\qquad \text{``00000110''}, \\
&\qquad stream\_content\_ext, \\
&\qquad stream\_content\_and\_component\_type, \\
&\qquad component\_tag, \\
&\qquad ISO\_639\_language\_code; \\
stream\_content\_ext \;&=\; 4 * binary\_digit; \\
stream\_content\_and\_component\_type \;&=\; \text{``000100000000''} \\
&\qquad \mid (\, \text{``0000''},\ component\_type\,); \\
component\_type \;&=\; 8 * binary\_digit; \\
binary\_digit \;&=\; \text{``0''} \mid \text{``1''}.
\end{aligned}
\tag{2}
$$

Here, we aim to create a descriptor with a composition that includes values reserved for future use related to the same problem described in Section 3.5.3.

In addition, as mentioned, there should be some restrictions for making the whole approach feasible and practical, as SBTVD presents a myriad of information elements with different uses and purely random data over a given structure, without considering the way it is processed, its natural limitations, and an entire TS and structure interrelation can be unacceptably longer and inefficient. For instance, the advanced audio coding (AAC) descriptor, in PMT sections, could have its parameter *profile_and_level* set with the wrong values [24, 67]. However, most DTV receivers usually ignore them, as seen in commercial platforms, and use the *stream_type* information in PMT sections to load the correct decoding library. Moreover, random values beyond the informed limits may also lead to no new condition, as data in TSs occupy a predefined bitwidth, and truncation often takes place. Consequently, no incapacitating problem would probably be revealed, and the resulting tests would be of no or little use. Finally, this understanding leads to another clue: some guidance or restriction during the related fuzzing strategy to tackle what is likely to reveal an existing problem.

Nonetheless, information related to previously identified problems, as presented in Section 3, parameters configured through GUIs of commercial equipment, and critical data with great adjustment flexibility (e.g., table descriptors) represent three crucial groups and recurrent causes of non-compliance. Those three aspects drive the intended grammar-based guided fuzzing approach and provide at least the initial coverage for creating tests. This way, those groups should be used as a basis for test creation since they have the potential to reveal fragility in receivers, as described in Section 3.

Fig. 6 illustrates the proposed test-generation methodology with the three mentioned groups: field problems, parameters configured in commercial equipment, and critical data with high flexibility. It summarizes our approach, where problems and grammar are sent to a fuzzing machine and then used to generate robustness tests in the three mentioned groups. The latter covers part of the universe defined by the target standards, which can be increased with new data fed to the mentioned fuzzing machine. Firstly, suppose there exist other data related to a field problem. In that case, it is likely that errors in it also result in similar problems that can be identified in audio and video interfaces, with the help of the techniques exposed in Section 5.1, which was already realized during the analysis of the real field problems in Section 3. This way, data related to a previous problem may also cause malfunction due to shared structures and incorrect or fragile code, which could be considered an "error region" around an original problem. The more one expands that area, the deeper possibly fragile structures are accessed, with a significant probability of incorrect operation (see the left part of Fig. 6. Besides, technicians' parameters configured in GUIs are also sources of errors since DTV standards are extensive, and no double-check is usually performed in receivers. Once again, another "error region" can be created around a single configuration available in a given GUI, which is even more critical when there is no value restriction. Then, sensitive and highly-flexible data are also prone to errors, given that the slightest deviation may lead receivers' routines to unexpected results (see Section 3.5.3). Indeed, DTV standards are not clear on descriptors and sometimes refer to documents created by other systems. Consequently, implementations do not usually consider all possible setups, and fragile areas are often identified. Finally, as illustrated in Fig. 6, error creation, which leads to areas around field problems, sensitive data, and parameters configured in GUIs, is performed with fuzzing. New values are randomly generated with knowledge regarding those three aspects and applicable format specifications, as mentioned here, from scratch.

Therefore, "error regions" centered on errors classified in one of the three mentioned groups could be continuously expanded if the related random process continues. In addition, while new field problems are identified, new GUIs of commercial equipment are developed, and additional or previously non-used descriptors begin to be considered, new "error regions" could then be fuzzed out.

The field problem in Section 3.2.1 is an example of the first group: SEI messages that carry AFD, closed-caption, and bar data [70, 71]. Hence, a fuzzer could insert errors in structure identifiers (*user_identifier* and *user_structure*), number of CC elements (*cc_count*), location of bars in their specific structure (*bar_data*), and specific AFD information (e.g., *afd_data*) [71], thus creating an "error region". Moreover, other fields in NAL units could also be fuzzed, e.g.,

DOI: 10.1002/stvr

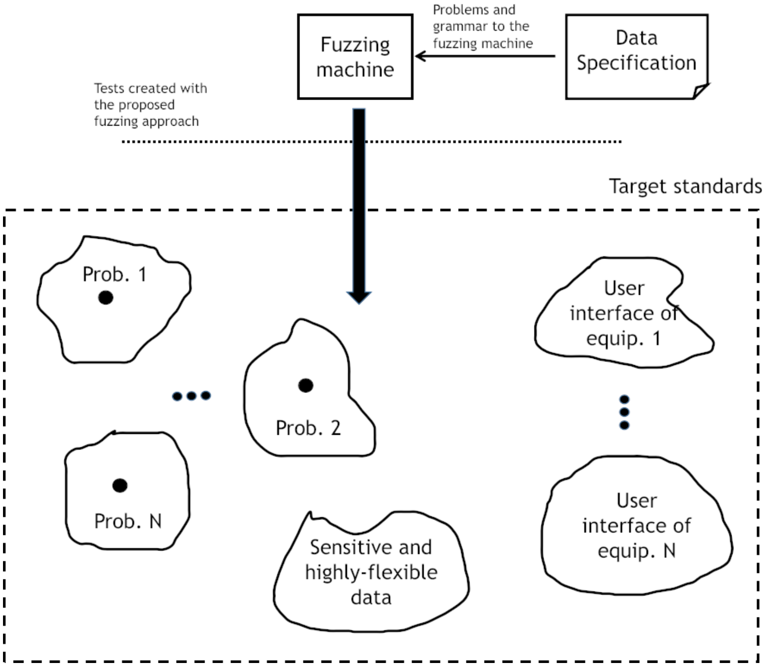
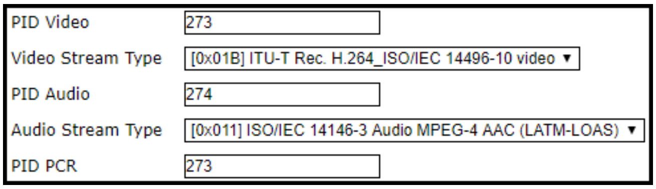

Fuzzing machine
Problems and grammar to the fuzzing machine
Data Specification
Tests created with the proposed fuzzing approach
Target standards
Prob. 1
Prob. 2
Prob. N
User interface of equip. 1
User interface of equip. N
Sensitive and highly-flexible data

Figure 6. The proposed strategy for test creation.

seq_parameter_set_rbsp() and *pic_parameter_set_rbsp()* [69]. Indeed, this approach has already been confirmed: TPV/Envision Brazil analyzed an occurrence in the same SEI messages, but now related to its marker bits [70, 71]. Besides, by that time, such a test had already been devised.

The second non-compliance group can be represented by the information sent in the field *stream_type* of PMT sections [23, 24], as in Section 3.5.1. It is usually configured in GUIs of head-end equipment, as illustrated in Fig. 7, which was reproduced from real equipment: *Video Stream Type* and *Audio Stream Type* can be freely set. The configured PID is recorded in the correct fields. However, no check regarding formats is performed. Other examples include application-profile information, such as *full-seg profile A* - FSA [85], and *component_tag* for media and object-carousel streams [20].



PID Video	273
Video Stream Type	[0x01B] ITU-T Rec. H.264_ISO/IEC 14496-10 video ▼
PID Audio	274
Audio Stream Type	[0x011] ISO/IEC 14146-3 Audio MPEG-4 AAC (LATM-LOAS) ▼
PID PCR	273

Figure 7. GUI of a commercial multiplexer.

Descriptors [24,67], which are dynamic information in PSI/SI tables [23,24], are natural members of the third group: sensitive data with extensive configuration. For instance, services can be informed in three distinct locations: PAT, SDT, and NIT, the latter providing classification (e.g., partial reception) and transmission parameters. Thus, since such data are used to store and classify services, fuzzing them may compromise device operation. Besides, services reported in those three tables should not present different numbers.

Following this methodology, the regions illustrated in Fig. 6 may grow around an initial problem, depending on evaluation needs. However, tests created through the mentioned steps are expected to cover many non-compliance types. Besides, such an evaluation system can be enhanced over time while new field problems are reported and sensitive data are tackled. Indeed, the proposed methodology gives rise to a system that is not born in its final form but evolves with time and can result in a comprehensive set of test cases. The following principles should also be considered:

- nonconformity tests in subsystem-oriented groups, such as PSI/SI; audio and video; and Ginga;
- only one non-compliance per TS so that the cause of malfunctions is isolated and identified;
- RF transmission is employed to use the reception chain.

Finally, an important notice: any evaluation should start later and end earlier, compared to a TS's life cycle, to avoid instability.

5.3. Tools for Video Analysis

Regarding video analysis, it is interesting to clarify some points. Video freezing, video flickering, artifacts, and frame skipping should be continuously checked with image processing algorithms during a given test. The necessary techniques for such tasks are simple and can be implemented through known techniques, such as histograms, correlation [72, 83], and structural similarity index (SSIM) evaluation [86, 87]. Indeed, the proposed image analysis algorithm employs, to some extent, freezing, flickering, artifacts, frame skipping identification, normalized cross-correlation, and SSIM. In summary, each algorithm's results are suitably combined to produce a robust merit figure. Finally, a camera must be in front of a TV display, which should be recognized to select only its screen, where video symptoms happen.

Consequently, a simple screen detection was devised based on known image processing algorithms and illustrated in Fig. 8. Initially, input images are re-scaled [83], given that the only goal is to detect a screen without specific object segmentation or more sophisticated evaluation. Then, some pre-processing is carried out, aiming at noise reduction [83]. Lastly, the developed algorithm tries to find a contour delimiting a screen, with the expected aspect ratio and shape, through edge detection [88] and corner identification, i.e., the latter being implemented through simple line intersection procedures. If that is the case for the current contour, it defines the screen area to be analyzed and, finally, tangential distortion is corrected [89], if present; otherwise, other contours are verified in search of four corners of a suitable rectangle, i.e., the one with correct aspect ratio and overall shape. Moreover, a camera should stay fixed in front of a TV display during a test sequence. Thus, once a TV screen is recognized, which happens only at the beginning of a test sequence, the same screen region is used for each nonconformity test until the last.

5.4. Tools for Audio Analysis

Once again, simple techniques can be used to provide detection of audio absence and discontinuities. In the present case, finite impulse response (FIR) filters [72] tuned to the chosen frequencies (i.e., 1 and 5KHz) can be used, whose outputs are continuously monitored through analysis of epochs of 20ms, which are sampled at 44.1 kHz. This way, no filter output during an entire test means absence, while periods of no output mean discontinuity. Besides, a more straightforward approach could be adopted based on audio amplitude, which undoubtedly requires noise floor calibration. Finally, in a more sophisticated analysis, tones distributed across the available frequency band could also be employed, as already mentioned.

5.5. The Complete Evaluation Procedure

The complete evaluation procedure includes receiver setup, signal transmission, channel acquisition, content decoding, video and audio evaluation, and report provision, as follows:

1. a camera and a microphone are correctly placed;
2. all devices are fixed and then the screen detection algorithm is run to define the respective screen's position;
3. a test sequence is loaded, along with the expected evaluation for each one, i.e., video and/or audio;

DOI: 10.1002/stvr

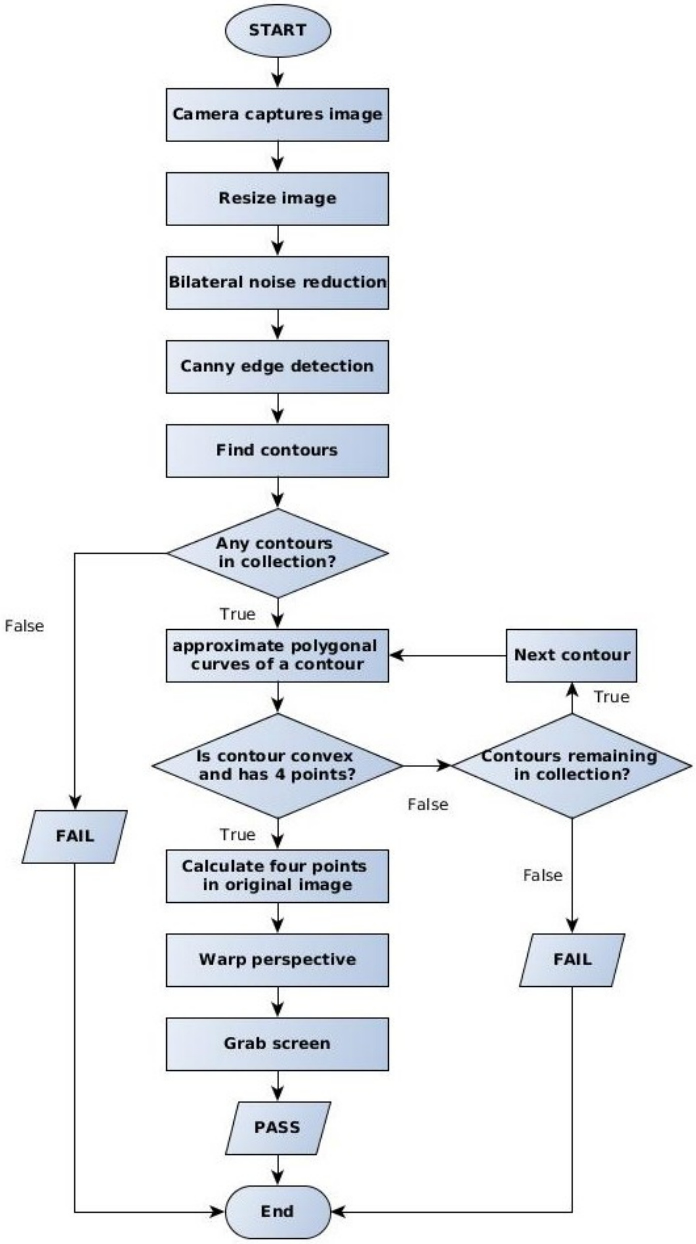

START
Camera captures image
Resize image
Bilateral noise reduction
Canny edge detection
Find contours
Any contours in collection?
False
True
approximate polygonal curves of a contour
Next contour
True
Is contour convex and has 4 points?
Contours remaining in collection?
False
FAIL
True
Calculate four points in original image
False
Warp perspective
FAIL
Grab screen
PASS
End

Figure 8. The algorithm developed for detecting a TV's screen.

4. the device is configured with infrared commands;
5. a dummy transmission is sent to allow a DTV receiver to scan and store the intended channel;
6. a specific test, with fuzzed data, is transmitted and the requested evaluation procedures are performed during the entire transmission;
7. the corresponding results are presented and stored;
8. steps 6 and 7 are repeated, until the last test;
9. after the last test, a complete report is issued, informing failure or success, along with results for each evaluation.

DOI: 10.1002/stvr

Finally, it is worth noticing that, in our approach, we first perform fuzzing and then store the resulting (broken) TSs for transmission. Although on-the-fly TS creation is possible, online multiplexing may primarily increase testing periods with no readily identifiable gain. In addition, TSs should also be available during correction and analysis phases as resources employed for error identification and handling.

6. EXPERIMENTAL EVALUATION

In order to evaluate the proposed methodology, a complete implementation was carried out, including testing environment, generation of non-conformant TSs (fuzzing), and experiment execution. In addition, to the best of the authors' knowledge, this is the first tool based on fuzzing that was developed for verifying non-conformance in DTV receivers. Moreover, it aims at noncompliance and configuration errors, which were never tackled. Consequently, there are no candidates we can directly compare with, in terms of performance, and no recent studies provide a clear research direction. We could even provide a simple comparison regarding fuzzing machines, using a general-purpose tool. However, with the present work, we hope to motivate more studies focused on robustness in DTV platforms and related networks.

In this regard, we are mainly interested in the following research questions:

RQ1 **(application feasibility)** Can the proposed evaluation methodology be used in practical evaluation processes and scenarios?

RQ2 **(commercial platform evaluation)** Can the proposed evaluation methodology provide a profile regarding the current installed-receivers base?

Regarding RQ1, we are interested in determining if the proposed methodology can be applied to real devices, which can be unfolded into two aspects: feasibility related to the evaluation of commercial platforms and its effectiveness. Besides, RQ2 is related to its result towards broadcasters, i.e., if the condition of the available commercial platforms can be accessed and considered during head-end configuration.

6.1. Description of the Non-Compliance Test Tool

The proposed methodology was implemented in *Python* and C++, running on a PC with Ubuntu 16.04. The resulting software can be divided into two parts: TS generation, according to Section 5.2, and automated test tool, which implements the control software illustrated in Fig. 4.

The multiplexing of non-compliant TSs was implemented with OpenCaster [90], where PSI/SI tables and descriptors were coded (and fuzzed) in *Python*. As a result, their fields can be modified according to our fuzzing-based test-creation strategy. Besides, structures in audio and video ESs were fuzzed, including wrong profiles, levels, and descriptions regarding several entities, such as LATM audio element headers [74] and H.264 NAL unit headers and slice types [69]. Moreover, PCR and PTS information and PES headers were also fuzzed. Next, audio, video, synchronization information, interactive application, and general data were multiplexed. One may notice that tests involve fuzzed Ginga applications and related structures, including digital storage media command and control (DSM-CC) carousels and associated signaling. Three categories were created: PSI/SI tables, audio and video, and Ginga, with tests according to the three groups mentioned in Section 5.

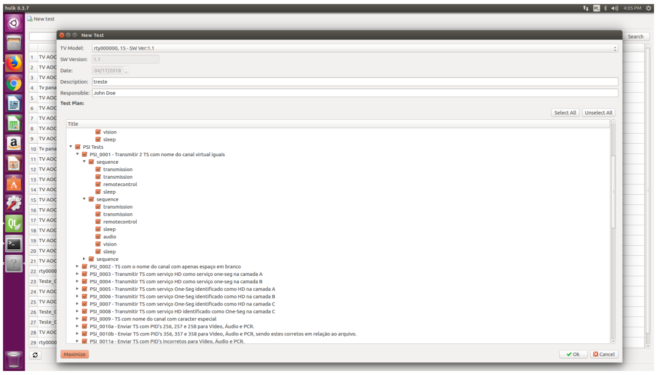
The complete evaluation tool is composed of modules that implement the proposed methodology. The image processing module is responsible for detecting and isolating a TV screen captured by a webcam, which records video at approximately 60 frames per second (FPS) and analyzes freezing, flickering, artifacts, and frame skipping. Freezing is detected using SSIM [86] to recognize frame sequences with a correlation larger than 95%. Flickering detection uses a combination of SSIM and normalized cross-correlation, while artifacts and frame skipping expect correlation in the range $50 - 70\%$ and lower than 50%, i.e., $0.5 \leq R_i \leq 0.7$ and $R_i < 0.5$, respectively. Finally, all evaluation algorithms were implemented with OpenCV [91] and some proprietary approaches specifically developed for them.

DOI: 10.1002/stvr

The transmission module was implemented with two devices DekTec DTU-215 [92] to emulate channel change, channel-associated information, and adjacent-channel reception. The remote control module uses an infrared transmitter managed by the Linux infrared remote control (LIRC) [93].

The audio-analysis module uses FIR filters tuned to 1 and 5 kHz and amplitude-based algorithms. Audio is captured with the advanced sound Linux architecture (ALSA) and then used for checking discontinuity and absence. The duration of each TS was empirically standardized in 30s, from which the first and the last 3s are discarded (cf. Section 5). Test cases are described in extensible markup language (XML) files [9] and fed to the developed tool.

The evaluation tool includes a configuration, control, and test-sequence monitoring GUI. There are screens for test configuration, test-sequence monitoring, and final result presentation. As an operator may only be interested in executing a subset of the available tests, a configuration screen was designed, where each test case can be enabled or disabled, as illustrated in Fig. 9. Indeed, tests or parts of them (e.g., audio and video analysis) can be selectively activated or not, using the same GUI.



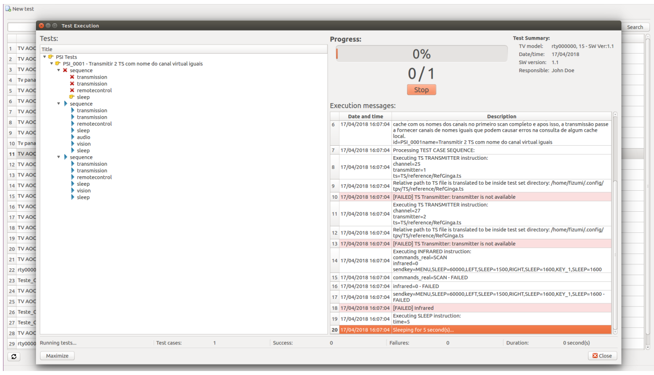
TV Model: rty000000, 15 - SW Ver:1.1
SW Version: 1.1
Date: 04/17/2018
Description: treste
Responsible: John Doe

Test Plan:

```
Title
      ☑ vision
      ☑ sleep
▾ ☑ PSI Tests
  ▾ ☑ PSI_0001 - Transmitir 2 TS com nome do canal virtual iguais
    ▾ ☑ sequence
        ☑ transmission
        ☑ transmission
        ☑ remotecontrol
        ☑ sleep
    ▾ ☑ sequence
        ☑ transmission
        ☑ transmission
        ☑ remotecontrol
        ☑ sleep
        ☑ audio
        ☑ vision
        ☑ sleep
    ▸ ☑ sequence
  ▸ ☑ PSI_0002 - TS com o nome do canal com apenas espaço em branco
  ▸ ☑ PSI_0003 - Transmitir TS com serviço HD como serviço one-seg na camada A
  ▸ ☑ PSI_0004 - Transmitir TS com serviço HD como serviço one-seg na camada B
  ▸ ☑ PSI_0005 - Transmitir TS com serviço One-Seg identificado como HD na camada A
  ▸ ☑ PSI_0006 - Transmitir TS com serviço One-Seg identificado como HD na camada B
  ▸ ☑ PSI_0007 - Transmitir TS com serviço One-Seg identificado como HD na camada C
  ▸ ☑ PSI_0008 - Transmitir TS com serviço HD identificado como One-Seg na camada C
  ▸ ☑ PSI_0009 - TS com nome do canal com caracter especial
  ▸ ☑ PSI_0010a - Enviar TS com PID's 256, 257 e 258 para Vídeo, Áudio e PCR.
  ▸ ☑ PSI_0010b - Enviar TS com PID's 356, 357 e 358 para Vídeo, Áudio e PCR, sendo estes corretos em relação ao arquivo.
  ▸ ☑ PSI_0011a - Enviar TS com PID's incorretos para Vídeo, Áudio e PCR.
```

Figure 9. Screen for selecting specific tests to be executed.

Finally, Figs. 10 and 11 show snapshots of two other screens. The first identifies the tests being run and the status of a test sequence. For instance, if some crash or major problem occurs, an in-depth verification of the corresponding log file is required to isolate a group of suspect TSs. If, after a specific test, all subsequent results show aberrant behavior, it should indicate the boundary of a test sequence for reproducing an incapacitating event. The second shows all data stored in a log file, which can further analyze a faulty execution.



Tests:

```
Title
▾ PSI Tests
  ▾ PSI_0001 - Transmitir 2 TS com nome do canal virtual iguais
    ▾ ✗ sequence
        ✗ transmission
        ✗ transmission
        ✗ remotecontrol
        sleep
    ▾ ▸ sequence
        ▸ transmission
        ▸ transmission
        ▸ remotecontrol
        ▸ sleep
        ▸ audio
        ▸ vision
        ▸ sleep
    ▾ ▸ sequence
        ▸ transmission
        ▸ transmission
        ▸ remotecontrol
        ▸ sleep
        ▸ vision
        ▸ sleep
```

Progress: 0% 0 / 1

Test Summary:
TV model: rty000000, 15 - SW Ver:1.1
Date/time: 17/04/2018
SW version: 1.1
Responsible: John Doe

Execution messages:

	Date and time	Description
6	17/04/2018 16:07:04	cache com os nomes dos canais no primeiro scan completo e apos isso, a transmissão passe a fornecer canais de nomes iguais que podem causar erros na consulta de algum cache local. id=PSI_0001name=Transmitir 2 TS com nome do canal virtual iguais
7	17/04/2018 16:07:04	Processing TEST CASE SEQUENCE:

8	17/04/2018 16:07:04	Executing TS TRANSMITTER instruction:
		channel=25
		transmitter=1
		ts=TS/reference/RefGinga.ts

9	17/04/2018 16:07:04	Relative path to TS file is translated to be inside test set directory: /home/fizumi/.config/tpv/TS/reference/RefGinga.ts
10	17/04/2018 16:07:04	[FAILED] TS Transmitter: transmitter is not available

11	17/04/2018 16:07:04	Executing TS TRANSMITTER instruction:
		channel=27
		transmitter=2
		ts=TS/reference/RefGinga.ts

12	17/04/2018 16:07:04	Relative path to TS file is translated to be inside test set directory: /home/fizumi/.config/tpv/TS/reference/RefGinga.ts
13	17/04/2018 16:07:04	[FAILED] TS Transmitter: transmitter is not available

14	17/04/2018 16:07:04	Executing INFRARED instruction:
		commands_real=SCAN
		infrared=0
		sendkey=MENU,SLEEP=60000,LEFT,SLEEP=1500,RIGHT,SLEEP=1600,KEY_1,SLEEP=1600

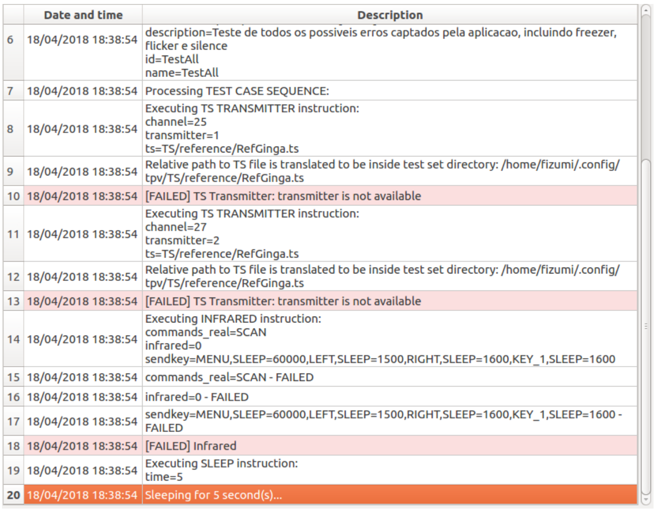
15	17/04/2018 16:07:04	commands_real=SCAN - FAILED
16	17/04/2018 16:07:04	infrared=0 - FAILED
17	17/04/2018 16:07:04	sendkey=MENU,SLEEP=60000,LEFT,SLEEP=1500,RIGHT,SLEEP=1600,KEY_1,SLEEP=1600 - FAILED
18	17/04/2018 16:07:04	[FAILED] Infrared
19	17/04/2018 16:07:04	Executing SLEEP instruction: time=5
20	17/04/2018 16:07:04	Sleeping for 5 second(s)...

Running tests... Test cases: 1 Success: 0 Failures: 0 Duration: 0 second(s)

Figure 10. The non-nompliance test tool executing tests.

The proposed evaluation methodology was implemented to assess and apply to real DTV receivers. Moreover, TPV/Envision Brazil will start using this tool to verify its receivers and provide better and more robust devices to the Brazilian market.

DOI: 10.1002/stvr



	Date and time	Description

6	18/04/2018 18:38:54	description=Teste de todos os possiveis erros captados pela aplicacao, incluindo freezer, flicker e silence
		id=TestAll
		name=TestAll

7	18/04/2018 18:38:54	Processing TEST CASE SEQUENCE:

8	18/04/2018 18:38:54	Executing TS TRANSMITTER instruction:
		channel=25
		transmitter=1
		ts=TS/reference/RefGinga.ts

9	18/04/2018 18:38:54	Relative path to TS file is translated to be inside test set directory: /home/fizumi/.config/tpv/TS/reference/RefGinga.ts
10	18/04/2018 18:38:54	[FAILED] TS Transmitter: transmitter is not available

11	18/04/2018 18:38:54	Executing TS TRANSMITTER instruction:
		channel=27
		transmitter=2
		ts=TS/reference/RefGinga.ts

12	18/04/2018 18:38:54	Relative path to TS file is translated to be inside test set directory: /home/fizumi/.config/tpv/TS/reference/RefGinga.ts
13	18/04/2018 18:38:54	[FAILED] TS Transmitter: transmitter is not available

14	18/04/2018 18:38:54	Executing INFRARED instruction:
		commands_real=SCAN
		infrared=0
		sendkey=MENU,SLEEP=60000,LEFT,SLEEP=1500,RIGHT,SLEEP=1600,KEY_1,SLEEP=1600

15	18/04/2018 18:38:54	commands_real=SCAN - FAILED
16	18/04/2018 18:38:54	infrared=0 - FAILED
17	18/04/2018 18:38:54	sendkey=MENU,SLEEP=60000,LEFT,SLEEP=1500,RIGHT,SLEEP=1600,KEY_1,SLEEP=1600 - FAILED
18	18/04/2018 18:38:54	[FAILED] Infrared
19	18/04/2018 18:38:54	Executing SLEEP instruction: time=5
20	18/04/2018 18:38:54	Sleeping for 5 second(s)...

Figure 11. Log screen shown at the end of a test-sequence.

6.2. Objectives

The performed evaluation used the implementation described in Section 6.1, with the following experimental goals:

EG1 **(application feasibility)** Show that the proposed methodology can be applied to commercial platforms, without massive effort for test execution and analysis.

EG2 **(commercial platform evaluation)** Provide an overview of commercial platforms available in the Brazilian market through the assessment of popular models from major multinational and national manufacturers.

EG3 **(methodology assessment)** Show that the proposed methodology can submit receivers to the mentioned scenarios and consequently reveal fragile implementations.

It is worth noticing that such experimental goals are closely related to the research questions raised at the beginning of Section 6. Specifically, EG1 and EG3 intend to answer RQ1, while EG2 is directly derived from RQ2.

6.3. Test Results

Experiments with our tool were performed using seven receivers from five different manufacturers, aiming at EG2. *Platform* 3 was released in 2013, *Platform* 2 in 2016, *Platforms* 1, 6, and 7 in 2017, and, finally, *Platform* 4 in 2020. The manufacturers responsible for such platforms are leaders in Brazil and represent 80% of its TV market. The environment illustrated in Fig. 4 was assembled, and all steps described in Section 5.5 were performed. Test results are expressed in Table II, where three categories can be identified: PSI and SI tables (PSI/SI); audio and video (A/V); and Ginga applications (Ginga). The configuration aspects tackled by each category are informed in Table III. The first four and the last two platforms are off-the-shelf ones; however, *Platform* 5 was still under development. Such a compound intends to provide a comparison among mature and in-development platforms.

Regarding A/V, *Platform* 2 presented a fail rate of 100%, while the others provided between 10% and 17.45%, which includes errors in audio and video ESs. For instance, there are H.264 streams with profile *high* and level 4.0, in their headers, although being encoded with *high* 10@5.0 [69]. Besides, there are many other tests regarding SEI, wrong structures, incorrect sequence, and picture headers [69, 70]. Audio streams also contain errors, such as the wrong number of channels, frequencies, and even compression tools (e.g., incorrect indication of spectral band replication) [74].

	Category	Total	Success	Fail	Percent Failed (%)
Platform 1	PSI/SI	556	555	1	0.18
	A/V	149	134	15	10.07
	Ginga	197	193	4	2.03
	All	**902**	**882**	**20**	**2.22**
Platform 2	PSI/SI	556	385	171	30.76
	A/V	149	0	149	100.00
	Ginga	197	195	2	1.02
	All	**902**	**580**	**322**	**35.70**
Platform 3	PSI/SI	556	397	159	28.6
	A/V	149	132	17	11.41
	Ginga	197	197	0	0.00
	All	**902**	**726**	**176**	**19.52**
Platform 4	PSI/SI	556	554	2	0.36
	A/V	149	131	18	12.08
	Ginga	197	196	1	0.51
	All	**902**	**881**	**21**	**2.33**
Platform 5	PSI/SI	556	495	61	10.97
	A/V	149	131	18	12.08
	Ginga	197	188	9	4.57
	All	**902**	**814**	**88**	**9.76**
Platform 6	PSI/SI	556	472	84	15.11
	A/V	149	133	16	10.74
	Ginga	197	139	58	29.44
	All	**902**	**744**	**158**	**17.51**
Platform 7	PSI/SI	556	340	216	38.85
	A/V	149	123	26	17.45
	Ginga	197	153	44	22.34
	All	**902**	**616**	**286**	**31.71**

Table II. Test results for an implementation of the proposed methodology, regarding seven different DTV platforms.

Such a result for $Platform$ 2, produced by a high-quality multinational manufacturer, shows its fragility, which is particularly important when faulty commercial equipment is used. One may also argue that some early tests may have caused this, but that was not the case. $Platform$ 7 is also produced by a multinational manufacturer, but it is still different from what was provided by the

Category	Test groups
PSI/SI	Tables PAT, PMT, NIT, SDT, CAT and EIT, together with their respective descriptors
	Table repeat periods
	Correlated fields
	Services
	Media encoding declarations
	PID declarations
	Table section control data
	Virtual channels
	Synchronization data
A/V	Video stream syntax
	Video stream syntax
	AAC stream elements
	LATM stream elements
	H.264 profiles and levels
	H.264 headers and parameter sets
	Audio specific elements (e.g., number of channels, sampling frequency, etc.)
	H.264 SEI messages
	H.264 frame information
	Video specific elements (e.g., frame rate, etc.)
Ginga	DSM-CC syntax
	DSM-CC descriptors
	DSM-CC compression
	DSM-CC Section control data
	Ginga application syntax
	Ginga APIs

Table III. Test groups included in each category.

others, i.e., an average of 11.28%. Moreover, manufacturers of *Platforms* 2 and 7 represent around 7% of the entire Brazilian TV market.

Platform 2 expects flawless ESs, which are not very common in real environments. The other platforms' failure rates are somewhat expected because some inherent robustness in DTV-device development usually exists. Nonetheless, there still exists room for improvement, mainly related to the most severe tests. For instance, platforms often anticipate audio issues, but problems associated with H.264 are usually left aside. Finally, even *Platform* 5, still under development, presented a failure rate for A/V of 12.08%, which can be explained with code reuse. Another significant result regards PSI/SI tests, including table parsing, descriptor decoding, and data classification and use. Indeed, source code related to PSI/SI tables is deeply revised due to its frequent use. Besides, this module is usually the primary target when a platform is customized for a DTV system. In that context, *Platforms* 1 and 4 presented low failure rates, which should be expected as most field problems are related to PSI/SI tables and usually lead to constant revision. $Platform$ 5 presented a higher rate, which is also expected, as it is a new model still under construction and did not use mature code. Moreover, this model presents enhancements and uses new libraries, which must still be tested and revised.

$Platforms$ 2, 3, 6, and 7 surprisingly presented fragile code, even being manufactured by companies with a long history in DTV. $Platform$ 2 is a model from 2016, while $Platforms$ 6 and 7 were released in 2017 and will probably be used for at least six more years. $Platform$ 3 was released in 2013, will probably be used for four more years, and is no longer sold. Thus, their impact seems severe, mainly because $Platform$ 3's fragility may be present in other models, and its manufacturer holds a market share of around 36.6%. Besides, $Platform$ 7, which presented the highest failure rate, is from a manufacturer with around 3% of market share. In summary, such tests reveal that if a DTV receiver project is not carefully executed and inherently robust, it may suffer from wrong data in head-end equipment and compromise DTV networks due to changes in transmission and new operators with sloppily configured equipment.

Finally, the lowest average failure rates regard interactive applications, which indicates a lot of development effort. Indeed, that is expected because Ginga-related standards are developed by the Fórum SBTVD, a committee dedicated to DTV standardization in Brazil. Moreover, this part of the SBTVD's standards was enhanced over the last decade, which usually triggers constant review and assessment. The highest failure rates were obtained with $Platforms$ 5, 6, and 7, the former also going through Ginga porting. That means Ginga is being interfaced with that platform's application programming interface, libraries, hypertext markup language (HTML) version 5.0 resident engine, and zapper module (the receiver-controlling application), which usually causes temporary instability. Consequently, it is expected that the same robustness level of $Platform$ 4, which is produced by the same manufacturer, is ultimately achieved. Nonetheless, the other two are commercial platforms, which may seem surprising. A possible explanation is a considerable change in the system and software stack, as is the case with $Platform$ 6, which may have caused undetected non-conformity. Besides, Ginga standards had just gone a review at that time, leaving many features unclear.

Although more sophisticated algorithms were developed and employed, amplitude detection was enough for audio malfunctions. Besides, all video artifacts and frame skipping events were also detected as flickering, which means this analysis would cover all the identified problems together with the freezing one. Consequently, audio amplitude and video freezing, and flickering reports may be enough.

As one can notice from Table II, a total of 902 tests were performed, which is not feasible with a manual approach. Indeed, on average, those 902 tests took 7.57 hours: 3.35 for Ginga, 0.9 for A/V, and 3.32 for PSI/SI, with the proposed methodology. Moreover, no human operator was required, which resulted in time for development and code enhancement while waiting for new evaluation results and fulfills EG1.

In addition, the need for software testing professionals is reduced, as one only needs to prepare the test environment and run a test sequence. Moreover, professional test suites usually provide thousands of tests, which indicates that the 902 tests presented here are just an initial set.

The chosen platforms are popular devices produced by market leaders, where software reuse is standard practice. Thus, a conclusion related to EG2 is that receivers are affected mainly by non-conforming data related to PSI/SI tables and media ESs, which should be taken into account by broadcasters during equipment configuration, thus guiding review and consistency checking. Therefore, any test round followed by "fix" phases should involve those groups, with fallback procedures and handling code also targeted on them.

In order to provide a complete picture, we have chosen two tests for further analysis. In the first, which was part of the PSI/SI category, a service of type *reserved* [24], i.e., $0x4A$ when fuzzed, was transmitted, which was identified with video freezing. In this case, receivers restricted processing and storage to services correctly classified in the service descriptor [24] in SDT sections, even though the field *program_number* in PAT and PMT was correctly configured. This problem affected almost all test platforms, which is of paramount importance.

The second one consisted of a Ginga application in a compressed carousel with a broken (fuzzed) *compression type descriptor* [76], which is very similar to the one in Section 3.6.1. As a result, this application was not loaded in most test platforms since the NCL entry point was not found. Only $Platform$ 1 was able to deal with it, which is very concerning.

Regarding the overall test results, Table II shows that $Platforms$ 1, 2, 3, 4, 5, 6, and 7 presented failure rates of 2.22, 35.70, 19.52, 2.33, 9.76, 17.51, and 31.71, respectively. It indicates that $Platforms$ 1 and 4 presented the best overall results and should keep stable during DTV reception in Brazil, but they still need improvements. Enhancements are now being performed in TPV/Envision Brazil's DTV receivers to reduce field-problem occurrence, which fulfills EG3.

As already mentioned, there is no similar tool we can use in comparison with our approach; However, in an attempt to fill this gap, we decided to compare only fuzzing machines. Indeed, even that is difficult to perform as most fuzzing tools freely available are focused on specific contexts, such as network, operating systems, or drivers [39]. In that sense, general-purpose ones seem viable. Among them, we were able to identify Peach [39], which is well known, effective, and flexible. However, it is also time-consuming to build its input format file using the indicated syntax. As one may notice, entire tools can even be built over Peach as long as the necessary definitions are provided. In our case, we have created the syntax of PMT sections, as shown in Table I, and then manually built ninety-five associated PMT sections with fuzzed data, which were later manually integrated into TSs. In contrast, we have also generated ninety-five test TSs with the tool developed here, based on the proposed methodology, and fed both sets to $Platform$ 2 and $Platform$ 3, resulting in Table IV. The latter presents columns with meanings similar to what is found in Table II. Those platforms were chosen based three reasons: they are popular models from multinational manufacturers, presented many problems, and are provided by market leaders. As one can notice, from Table IV, TSs generated with the proposed methodology helped uncover more fragile parts than Peach. Indeed, that is expected due to the test-creation strategy adopted here, which is focused on the most likely ways problems occur. Again, this comparison considers only the underlying fuzzing machines, with favorable results for the proposed methodology.

	The proposed Methodology				**Peach [39]**			
	Total	**Success**	**Fail**	**P. Failed (%)**	**Total**	**Success**	**Fail**	**P. Failed (%)**
Platform 2	95	59	36	37.89%	95	75	20	21.05%
Platform 3	95	55	40	42.11%	95	82	13	13.68%

Table IV. Tests comparing problems uncovered with fuzzed data created with the proposed methodology and Peach [39].

It is worth mentioning that the results provided here can be promptly used for improving the evaluated platforms. Indeed, when a product is being developed, it is submitted to laboratory-created and few real DTV signals (TSs) recorded in the field (e.g., user premises, public spaces, etc.), whose content may reveal a problem during device operation. Consequently, software fixes involve two primary artifacts: problem description and associated TS, which are the same as the proposed methodology, and problem description comes from the identified behavior in the audio or video output. However, such TSs recorded in the field are not numerous and usually contain only what is expected and dictated by standards, with no purposefully incorrect data. Consequently, as the latter is the very focus of this work, a new aspect regarding a receiver's processing chain is then tackled, and that has the potential to reveal completely unexpected behaviors.

It is also important to clarify that the proposed methodology aims to reveal fragility. That being said, we know in advance that the created TSs bear non-conforming data and are wrong from the viewpoint of what is dictated by the associated standards. This way, the target is to verify if a given receiver is ready to handle such problems due to a previous fallback procedure, unintended consequences, restricted state spaces, or will present abnormal behavior. Moreover, all results presented here were manually confirmed, which, during our initial tests, revealed some false negatives from video evaluation but no false positives at all. Later, a brief analysis showed the need for adjustments in some correlation thresholds and the standard test duration, avoiding those occurrences. Although that is true for the current composition of the developed tool, new checks or evaluation groups may cause additional occurrences as other symptoms that do not taper down towards the known ones may happen.

Another fair question regards cost: is a new testing tool worthwhile? To answer that, some relative figures are required. In TPV/Envision, we have isolated the average yearly cost relative to the kind of field problem presented in Section 3, as provided by the after-sales department, which is around 20% of its total amount, the latter in the order of a few million dollars. Moreover, this value, around one million dollars, is higher than what was invested in developing the proposed tool. Consequently, the return on investment can come already in the first year of its use.

One question remains of how broad the evaluation performed here is, which is closely related to EG2. Indeed, the Brazilian market provides much more than the seven devices employed here. Nonetheless, although only those platforms were employed, the chosen manufacturers represent five out of the first seven market leaders and amount to 80% of the Brazilian market. Moreover, software reuse is usual, even throughout different models and brands from the same manufacturer, given that a platform maybe even employed for low-, mid-, and high-end products. Consequently, the rough estimates are reasonable, although some discrepancies may be noticed.

7. RELATED WORK

As mentioned in Section 6, the present work seems to be the first study to tackle robustness assessment regarding DTV receivers, with fuzzed TSs that aim to discover fragile parts in their processing chains. Indeed, the approach presented here, i.e., recognizing that configuration errors take place and that they should be anticipated so that non-conformance is purposefully inserted to evaluate and handle their consequences, is novel and has never been reported in the available literature. Consequently, it is not feasible to compare it with other studies, to define its performance relative to what currently exists. The majority of the current studies on DTV evaluation focus on conformance testing, which is usually spread across different DTV subsystems. Even so, we will provide a review of TV testing in general while trying to identify some similarities with our work.

Tekcan *et al.* [94] proposed a black-box testing framework with automatic test-case generation, focusing on user interaction and features accessed via GUI. Unlike the scheme proposed here, non-conformance and configuration-error test cases are not addressed. Rau [95] tackled only image quality, while Belém *et al.* [96] restricted their method to evaluating general problems of finished elements in production lines. In turn, Souza Júnior *et al.* [32] developed a framework for automatic field-testing, focusing on receiver execution and user interaction when handling real signals. Again, although there are slight similarities when interactive applications are involved, their focus is entirely different. Park *et al.* [97], in turn, tackled different aspects: radiofrequency (RF), RF channels, and signal level.

The studies developed by Pinheiro *et al.* [15] and Souza Júnior *et al.* [21], together with the HbbTV test suite [98], for instance, tackle interactive applications and indeed present some similarity with the work proposed here. However, their focus is on conformity, as is the case with the work developed by Flores-Guridi *et al.* [99], which aims to provide an approval protocol for DTV receivers. Once again, no mention of transport layer, processing chain, robustness, or non-conformance was identified. Even so, Souza Júnior *et al.* [21] provided some results we can mention. They showed error-profile progression across scrum sprints, with an expected reduction in the corresponding number of bugs. Consequently, during development, it is clear that conformance assessments can easily reveal failure rates as high as 60%, which tends to be reduced over time towards 0, until a final software release is provided. In the case of the present paper, if only the test-group Ginga is considered, the highest found figure was 29.44%, which is surprising, given that $Platform$ 6 is a commercial one, and shows the importance of such an evaluation procedure integrated within development cycles.

In summary, there are studies focused on the three main DTV layers: transmission/physical, transport, and media, the latter including audio, video, multimedia, and interactive applications. However, no identifiable work tackles robustness and non-conformance with a focus on DTV networks or related systems with similar protocols (e.g., DR systems), as noticed from the associated transport layer. Furthermore, the underlying fuzzing machine is classified as a generation one and performs better than simple fuzzing. Consequently, the present work configures as a milestone regarding DTV-receiver robustness evaluation and tackles an aspect still disregarded: configuration errors originating from broadcasters.

8. CONCLUSIONS

This paper proposes a novel robustness testing methodology for evaluating DTV receivers based on non-compliance tests via fuzzing to identify opportunities for receiver improvement. In particular, it presents a collection of real field problems identified in DTV networks and outlines a scheme for non-compliance insertion that performs grammar-based guided fuzzing. The resources tackled here are available in most DTV standards worldwide, and some problems were identified in different DTV systems. Consequently, the findings presented here are novel to system and software test practitioners of DTV receivers and are of general and broad applicability.

The proposed methodology neither indicates test areas nor informs the number of evaluation procedures. Such decisions may result in different implementations for detecting specific problems and tackling completely different scenarios, which also depends on the fuzzing approach itself. Thus, its implementation consists of a system that must be continuously extended. Furthermore, the specific cause of a problem is not identified, as it detects only what is noticed by users and through the available video and audio interfaces. Thus, when a problem is informed, the next step is to analyze the respective TS and the associated receiver to develop a handling procedure.

The proposed methodology was implemented, and experiments allowed us to identify robustness problems related to wrong information in several DTV subsystems, such as media ESs and service classification. The most severe problems seem to be related to media ESs, which caused a failure rate of 100% in one test platform and PSI/SI data.

A significant result resides in the snapshot of the platforms provided by multinational manufacturers. Furthermore, given the software reuse practice, the chosen receivers were fragile and could be extrapolated for many devices and markets.

Another point regards attackers' intentional manipulation of transmitted configuration. Currently, that seems even more possible as modern systems may be primarily online, which is expected, for instance, in the context of pay-TV operators. In contrast, our work is entirely based on mistakes made during equipment configuration or development errors, and the raised matter seems somewhat out of scope. However, intentional errors can also be handled by the proposed methodology as long as they occur in the context of configuration data.

Although the present work was initially focused only on DTV networks, which may seem a restriction, there is a myriad of different broadcasting systems that can benefit from it, including DTV and DR ones and also satellite and cable counterparts, such as DVB-S and DVB-C. However, DVB-S and DVB-C are mainly used in vertical markets; that is, the related operators control both ends of DTV networks: they transmit signals and also provide receivers, the latter usually using similar underlying software systems (e.g., middleware, system software, etc.). Moreover, their middleware modules are usually similar or even the same across different devices, with a more homogeneous behavior. Consequently, they are responsible for their entire networks, a condition that ultimately speeds up correction procedures as they are highly committed to keeping a certain quality of service. DVB-T and terrestrial systems, in turn, are intended for horizontal markets, i.e., different broadcasters and a myriad of independent platform manufacturers, the latter having no relation to the former, making everything more complex and delaying or even impeding any action regarding configuration correction. This way, when a problem happens in terrestrial networks, a solution will certainly come much later than in satellite or cable ones, creating a more demanding environment regarding robustness against configuration errors. This understanding thus led the present work to evaluate terrestrial networks with an implementation of the proposed methodology. However, that does not exempt satellite and cable networks from benefiting from robust devices, and the framework developed here may also be of interest to them. Moreover, regarding configuration, robustness under real operation is not often tackled in many communication systems. In contrast, the present work shows a methodology whose essence can be applied to awaken interest in such a context. In addition, it is worth noticing that practical fuzzers usually target specific areas, such as network protocols, which, together with the simplex communication of DTV systems, led to some characteristics of the proposed methodology.

As future work, other problem groups will be developed, such as generic PES structure and CC. In addition, a generic non-conformance testing tool based on the proposed methodology will be made available to enhance the robustness of DTV networks. Moreover, enhancements to the underlying fuzzer will be provided, such as gradients and machine learning algorithms that provide adaptability towards known fragile parts and subsystems. Finally, given that TPV/Envision is a member of the SBTVD's technical committee, the provision of the tool developed here for the general public is being evaluated.

ACKNOWLEDGMENT

TPV/Envision funds the work in this paper.

REFERENCES

1. International Telecommunication Union. *Report ITU-R BT.2140-3 – Transition from analogue to digital terrestrial broadcasting* 2011.
2. Jack K. *Video Demystified: A Handbook for the Digital Engineer*. 5th edn., Newness, 2011.
3. Reimers U. Dvb-the family of international standards for digital video broadcasting. *Proc. of the IEEE* 2006; **94**(1):173–182, doi:10.1109/JPROC.2005.861004.
4. Majstrenko VA, Averchenko AP, Zhenatov BD. Dvb-t2 advantages and its construction features on the base of dvb-t equipment. *Proceedings of the 2012 IEEE 11th International Conference on Actual Problems of Electronics Instrument Engineering (APEIE)* Oct 2012; doi:10.1109/APEIE.2012.6629158.
5. Dumic E, Grgic S, Frank D. Simulating dvb-t to dvb-t2 migration opportunities in croatian tv broadcasting. *Proceedings of the 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* 2014; :1–5doi:10.1109/SOFTCOM.2014.7039121.
6. Advanced Television Systems Committee. *A/53: ATSC Digital Television Standard, Parts 1 - 6* Jan 2007. URL `https://www.atsc.org/wp-content/uploads/2015/03/a_53-Part-1-6-2007.pdf`.
7. Fay L. Atsc 3.0 physical layer overview. *Proceedings of the 2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting* Jun 2015; doi:10.1109/BMSB.2015.7177257.
8. Takada M, Saito M. Transmission system for isdb-t. *Proceedings of the IEEE* January 2006; **94**(1):251–256, doi: 10.1109/JPROC.2005.859692.
9. Farias B, Araújo N, Fabrício R, B da Costa J, de Lima Filho EB. A methodology for convergence between ginga and hbbtv. *Proceedings of the International Conference on Consumer Electronics (ICCE)* 2019; doi: 10.1109/ICCE.2019.8662049.
10. Morris S, Smith-Chaigneau A. *Interactive TV Standards: A Guide to MHP, OCAP, and JavaTV*. 1st edn., Focal Press, 2005.
11. Crinon RJ, Bhat D, Catapano D, Thomas G, Loo JTV, Bang G. Data broadcasting and interactive television. *Proceedings of the IEEE* January 2006; **94**(1):102–118, doi:10.1109/JPROC.2005.861020.
12. Advanced Television Systems Committee. *ATSC 3.0 Interactive Content* May 2019. URL `https://www.atsc.org/wp-content/uploads/2017/12/A344-2019-Interactive-Content-1.pdf`.
13. Farias MCQ, Alencar M, Carvalho MM. Digital television broadcasting in brazil. *IEEE Multimedia* may 2008; **15**(2):64–70, doi:10.1109/MMUL.2008.25.
14. *Ginga-NCL Conformance Testing*. Http://testsuite.gingancl.org.br/ [Online; accessed 10-December-2019].
15. Pinheiro CF, Eddie Filho B, Oliveira RR, Cavalcante AA, Klehm VS, Pereira DP, Melo HL. A conformity test-suite proposal for ginga-ncl. *Proceedings of the XXX Brazilian Symposium on Telecommunications (SBRT'12)* 2012; (in portuguese).
16. Soares LFG, Moreno MF, Neto CDSS, Moreno MF. Ginga-NCL: Declarative middleware for multimedia IPTV services. *IEEE Communications Magazine* June 2010; **48**(6):74–81, doi:10.1109/MCOM.2010.5473867.
17. de Souza Filho G, Cunha Leite L, Coelho Freire Batista C. Ginga-j: the procedural middleware for the brazilian digital tv system. *Journal of the Brazilian Computer Society* March 2007; **13**(1):47–56, doi:10.1007/BF03192401.
18. Soares LFG, Rodrigues RF, Moreno MF. Ginga-ncl: the declarative environment of the brazilian digital tv system. *Journal of the Brazilian Computer Society* March 2007; **13**(1):37–46, doi:10.1007/BF03192400.
19. Rech J, Freitas V, Farias B, de Lima Filho EB, B da Costa J, Machado I, Chen X, Pinheiro C, Xavier D. A methodology for providing encrypted-content decoding in dtv play. *Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE)* 2021; doi:10.1109/ICCE50685.2021.9427703.

20. Brazilian Association of Technical Standards. *ABNT NBR 15604, Digital terrestrial television - Receivers* 2007.
21. Souza Júnior MJ, Maia OB, de Lima Filho EB, Fabrício R, Silva A. An automated testing methodology for digital tv middleware implementations. *Proceedings of the 2019 IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)* 2019; .
22. Brazilian Association of Technical Standards. *ABNT NBR 15601-1, Digital terrestrial television - Transmission system* 2007.
23. International Organization for Standardization and International Electrotechnical Commission. *ISO/IEC 13818-1, Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Systems* 2007.
24. Brazilian Association of Technical Standards. *ABNT NBR 15603, Digital terrestrial television - Multiplexing and service information (SI)* 2015.
25. Brazilian Association of Technical Standards. *ABNT NBR 15602-1, Digital terrestrial television - Video coding, audio coding and multiplexing Part 1: Video coding* 2007.
26. Brazilian Association of Technical Standards. *ABNT NBR 15602-2, Digital terrestrial television - Video coding, audio coding and multiplexing Part 2: Audio coding* 2007.
27. Brazilian Association of Technical Standards. *ABNT NBR 15606-2, Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 2: Ginga-NCL for fixed and mobile receivers – XML application language for application coding* 2018.
28. Savino HJ, de Lima Filho EB. Program clock reference correction in transport stream processors with rate adaptation. *Multimedia Tools and Applications* Jun 2017; **76**(12):14 107–14 128, doi:10.1007/s11042-016-3814-3.
29. Miller BP, Fredriksen L, So B. An empirical study of the reliability of unix utilities. *Communications of the ACM* December 1990; **33**(12):32–44, doi:10.1145/96267.96279.
30. Sutton M, Greene A, Amini P. *Fuzzing: brute force vulnerability discovery*. 1st ed. edn., Addison-Wesley Professional, 2007.
31. Li J, Zhao B, Zhang C. Fuzzing: a survey. *Cybersecurity* 2018; **1**(6):1–13, doi:10.1186/s42400-018-0002-y.
32. Souza Júnior M, Maia OB, Leite SC, de Lima Filho EB, Izumi F, Andrade RR, Corrêa P. A framework for automatic field evaluation of dtv receivers. *Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE)* 2019; :1–5doi:10.1109/ICCE50685.2021.9427643.
33. Izumi F, Farias B, de Lima Filho E, Amorim A, Maia OB, Silva A. Evaluation of digital tv receivers with noncompliant mpeg-2 transport streams. *Proceedings of the IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)* 2019; :1–2doi:10.1109/ICCE-TW46550.2019.8991693.
34. Godefroid P, Kiezun A, Levin MY. Grammar-based whitebox fuzzing. *Proceedings of the ACM SIGPLAN 2008 Conference on Programming Language Design and Implementation* May 2008; doi:10.1145/1379022.1375607.
35. Pham VT, Böhme M, Roychoudhury A. Model-based whitebox fuzzing for program binaries. *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)* 2016; :543–553doi: 10.1145/2970276.2970316.
36. Kargén U, Shahmehri N. Turning programs against each other: high coverage fuzz-testing using binary-code mutation and dynamic slicing. *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE)* August 2015; doi:10.1145/2786805.2786844.
37. Wang P, Zhou X, Lu K, Yue T, Liu Y. The progress, challenges, and perspectives of directed greybox fuzzing. *ArXiv Preprint ArXiv:2005.11907* February 2021; :1–16.
38. Eberlein M, Noller Y, Vogel T, Grunske L. Evolutionary grammar-based fuzzing. *ArXiv Preprint arXiv:2008.01150* August 2020; :1–15.
39. Liang H, Pei X, Jia X, Shen W, Zhang J. Fuzzing: State of the art. *IEEE Transactions on Reliability* Sep 2018; **67**(3):1199 – 1218, doi:10.1109/TR.2018.2834476.
40. Pham VT, Böhme M, Roychoudhury A. Aflnet: A greybox fuzzer for network protocols. *Proceedings of the 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)* August 2020; doi: 10.1109/ICST46399.2020.00062.
41. Luo Z, adn Yuheng Shen FZ, Jiao X, Chang W, Jiang Y. Ics protocol fuzzing: Coverage guided packet crack and generation. *Proceedings of the 2020 2020 57th ACM/IEEE Design Automation Conference (DAC)* July 2020; doi: 10.1109/DAC18072.2020.9218603.
42. Gorbunov S, Rosenbloom A. Autofuzz: Automated network protocol fuzzing framework. *International Journal of Computer Science and Network Security* August 2010; **10**(8):239–245.
43. Abreu RB, Gadelha MYR, Cordeiro LC, de Lima Filho EB, da Silva Jr WS. Bounded model checking for fixed-point digital filters. *J. Braz. Comput. Soc.* 2016; **22**(1):1:1–1:20, doi:10.1186/s13173-016-0041-8.
44. Pereira PA, Albuquerque HF, da Silva I, Marques H, Monteiro FR, Ferreira R, Cordeiro LC. Smt-based context-bounded model checking for CUDA programs. *Concurr. Comput. Pract. Exp.* 2017; **29**(22), doi:10.1002/cpe.3934.
45. Dorofeeva R, El-Fakih K, Yevtushenko N. An improved conformance testing method. *Formal Techniques for Networked and Distributed Systems - FORTE 2005, 25th IFIP WG 6.1 International Conference, Taipei, Taiwan, October 2-5, 2005, Proceedings*, *Lecture Notes in Computer Science*, vol. 3731, Wang F (ed.), Springer, 2005; 204–218, doi:10.1007/11562436_16.
46. Monteiro FR, Garcia M, Cordeiro LC, de Lima Filho EB. Bounded model checking of C++ programs based on the qt cross-platform framework. *Softw. Test. Verification Reliab.* 2017; **27**(3), doi:10.1002/stvr.1632.
47. Fischer W. *Digital Video and Audio Broadcasting Technology: A Practical Engineering Guide*. 3rd edn., Springer, 2010.
48. Brazilian Association of Technical Standards. *ABNT NBR 15606-4, Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 4: Ginga-J – The environment for the execution of procedural applications* 2016.
49. Wu Y, Hirakawa S, Reimers U, Whitaker J. Overview of digital television development worldwide. *Proc. of the IEEE* Jan 2006; **94**(1):8–21, doi:10.1109/JPROC.2005.861000.
50. digi.TV. *Conformance Test Specification - Recommendations* Mar 2012. (Project South-East European Digital Television).

51. Rohde & Schwarz. *DTV: transmission perfect - transport stream correct?* 2010. URL https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_news_from_rs/200/N200_DTV-transmission-perfect_e.pdf.
52. European Telecommunications Standard Institute. *ETSI TR 101 290, Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems*. 1.2.1 edn. May 2001.
53. European Telecommunications Standard Institute. *TR 101 291, Digital Video Broadcasting (DVB); Usage of the DVB test and measurement signalling channel (PID 0x001D) embedded in an MPEG-2 Transport Stream (TS)*. 1.1.1 edn. Jun 1998. URL https://www.tek.com/datasheet/transport-stream-compliance-analyzer.
54. NorDig. *Rules of Operation of Service Information in the Finnish DTTV Networks*. 1.1 edn. Dec 2014.
55. NorDig. *NorDig unified requirements for integrated receiver decoders for use in cable satellite terrestrial and IP-based networks*. 2.2 edn. 2009.
56. European Telecommunications Standard Institute. *ETSI TR 101 154, Digital Video Broadcasting (DVB); Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting application*. 1.4.1 edn. Jul 2000. URL http://www.etsi.org/deliver/etsi_tr/101100_101199/101154/01.04.01_60/tr_101154v010401p.pdf.
57. European Telecommunications Standard Institute. *ETSI TR 101 190, Digital Video Broadcasting (DVB); Implementation guidelines for DVB terrestrial services; Transmission aspects*. 1.3.2 edn. May 2011. URL http://www.etsi.org/deliver/etsi_tr/101100_101199/101190/01.03.02_60/tr_101190v010302p.pdf.
58. Association of Radio Industries and Businesses. *ARIB TR-B14, Operational Guidelines for Digital Terrestrial Television Broadcasting*. 2.8-e2 edn. May 2006. URL https://www.arib.or.jp/english/html/overview/doc/8-TR-B14v2_8-1p3-1-E2.pdf.
59. Brazilian Association of Technical Standards. *ABNT NBR 15608-1, Digital terrestrial television – Operational guideline Part 1: Transmission system – Guideline for ABNT NBR 15601:2007 implementation* 2018.
60. Brazilian Association of Technical Standards. *ABNT NBR 15608-2, Digital terrestrial television – Operational guideline Part 2: Video coding, audio coding and multiplexing – Guideline for ABNT NBR 15602:2007 implementation* 2010.
61. Brazilian Association of Technical Standards. *ABNT NBR 15608-3, Digital terrestrial television – Operational guideline Part 3: Multiplexing and service information (SI) – Guideline for ABNT NBR 15603:2007 implementation* 2017.
62. EiTV. *EiTV Inspector*. Https://www.eitv.com.br/produtos/eitv-inspector/ [Online; accessed 10-December-2019].
63. Sayood K. *Introduction to Data Compression*. 5th edn., Morgan Kaufmann, 2017.
64. Fischer W. *Digital Television: A Practical Guide for Engineers*. Springer Science & Business Media, 2004.
65. European Telecommunications Standard Institute. *ETSI TR 101 202: Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting*. 1.2.1 edn. jan 2003. URL http://www.etsi.org/deliver/etsi_tr/101200_101299/101202/01.02.01_60/tr_101202v010201p.pdf.
66. Brazilian Association of Technical Standards. *ABNT NBR 15610-2, Digital terrestrial television - Accessibility Part 2: Sound functionalities* 2012.
67. European Telecommunications Standard Institute. *ETSI EN 300 468: Digital Video Broadcasting (DVB); Specification for Service Inform ation (SI) in DVB systems*. 1.15.1 edn. Mar 2016. URL http://www.etsi.org/deliver/etsi_en/300400_300499/300468/01.15.01_60/en_300468v011501p.pdf.
68. Reimers U. *DVB: The Family of International Standards for Digital Video Broadcasting*. 2nd edn., Springer-Verlag Berlin Heidelberg, 2004.
69. International Telecommunication Union Telecommunication Standardization Sector (ITU-T). *ITU-T Recommendation H.264, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services – Coding of moving video, Advanced video coding for generic audiovisual services* Mar 2005. URL https://www.itu.int/rec/T-REC-H.264-200503-S/en.
70. Advanced Television Systems Committee. *ATSC Standard A/72 Part 1 – Video System Characteristics of AVC in the ATSC Digital Television System* May 2015. URL https://www.atsc.org/wp-content/uploads/2015/03/A72-Part-1-2015.pdf.
71. Society of Cable Telecommunications Engineers. *ANSI/SCTE 128-1 2013, AVC Video Constraints for Cable Television, Part 1- Coding* 2013. URL https://www.scte.org/documents/pdf/Standards/ANSI_SCTE%20128-1%202013.pdf.
72. Diniz P, da Silva E, Netto S. *Digital Signal Processing: System Analysis and Design*. 2nd edn., Cambridge University Press, 2010. URL https://books.google.com.br/books?id=HoWaAgAAQBAJ.
73. International Organization for Standardization and International Electrotechnical Commission. *ISO/IEC 13818-7, Generic coding of moving pictures and associated audio information — Part 7: Advanced Audio Coding (AAC)* 2004.
74. International Organization for Standardization and International Electrotechnical Commission. *ISO/IEC 14496-3: Information technology –coding of audio-visual Objects – Part3: Audio* 2009. URL https://books.google.com.br/books?id=MAvRoAEACAAJ.
75. EiTV. *EiTV Playout Professional*. Https://www.eitv.com.br/produtos/eitv-playout-professional/ [Online; accessed 10-December-2019].
76. Brazilian Association of Technical Standards. *ABNT NBR 15606-3, Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 3: Data transmission specification* 2018.
77. Broek F, Hond B, Torres AC. Security testing of gsm implementations. *Proceedings of the 6th International Symposium on Engineering Secure Software and Systems* February 2014; **8364**:179–195, doi:10.1007/978-3-319-04897-0_12.
78. Liu M, Crussiere M, Helard JF, Pasquero OP. Analysis and performance comparison of dvb-t and dtmb systems for terrestrial digital tv. *Proceedings of the 2008 11th IEEE Singapore International Conference on Communication Systems* 2008; .
79. Zepernick HJ, Iqbal MI, Khatibi S. Quality of experience of digital multimedia broadcasting services: An experimental study. *Proceedings of the 2016 IEEE Sixth International Conference on Communications and*

Electronics (ICCE) 2016; .

80. Morello A, Mignone V. Dvb-s2: The second generation standard for satellite broad-band services. *Proceedings of the IEEE* January 2006; **94**(1):210–227, doi:10.1109/JPROC.2005.861013.
81. Robert J, Schaaf C, Stadelmeier L. Dvb-c2 - the standard for next generation digital cable transmission. *Proceedings of the 22009 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting* 2009; .
82. Wang KP, Wang ZG, Lei XM, Zhou JZ, Cao X, Zhang WR. Rf front-end ics for digital radio broadcasting drm and dab. *Proceedings of the 2009 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC)* 2009; .
83. Jain AK. *Fundamentals of Digital Image Processing*. 1st edn., Prentice-Hall: Upper Saddle River, NJ, USA, 1989.
84. International Organization for Standardization and International Electrotechnical Commission. *ISO/IEC 14977, Information technology - Syntactic metalanguage - Extended BNF* 1996.
85. Brazilian Association of Technical Standards. *ABNT NBR 15606-1, Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 1: Data coding specification* 2018.
86. Brunet D, Vrscay ER, Wang Z. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing* April 2012; **21**(4):1488–1499, doi:10.1109/TIP.2011.2173206.
87. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* April 2004; **13**(4):600–612, doi:10.1109/TIP.2003.819861.
88. Ma X, Li B, Zhang Y, Yan M. The canny edge detection and its improvement. *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence* 2012; :50–58doi:10.1007/978-3-642-33478-8_7.
89. Weng J, Cohen P, Herniou M. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis And Machine Intelligence* Oct 1992; **14**(10):965–980, doi:10.1109/34.159901.
90. Avalpa. *OpenCaster 3.2.2: the free digital tv software*. Http://www.avalpa.com/the-key-values/15-free-software/33-opencaster [Online; accessed 10-December-2019].
91. OpenCV Foundation. *Open Source Computer Vision Library (OpenCV)*. Https://opencv.org/ [Online; accessed 10-December-2019].
92. DekTec. *DTU-215: Cable/terrestrial modulator for USB-2*. Https://www.dektec.com/products/USB/DTU-215/ [Online; accessed 10-December-2019].
93. Karsten Scheibler & Christoph Bartelmus. *Linux Infrared Remote Control (LIRC)*. Http://www.lirc.org [Online; accessed 10-December-2019].
94. Tekcan T, Zlokolica V, Pekovic V, Teslic N, Gündüzalp M. User-driven automatic test-case generation for dtv/stb reliable functional verification. *IEEE Transactions on Consumer Electronics* may 2012; **58**(2):587–595, doi:10.1109/TCE.2012.6227464.
95. Rau A. Automated test system for digital tv receivers. *Proceedings of the International Conference on Consumer Electronics* 2000; :1–2doi:10.1109/ICCE.2000.854597.
96. Belém R, Cruz C, Kimura P, Amorim A, Filho EL, Silva O, Coimbra L. An architecture for test execution in video monitor and digital tv receiver production lines. *Proceedings of the 2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)* 2019; :1–2doi:10.1109/ICCE-TW46550.2019.8991991.
97. Park SI, Kim J, Choi D, Kim HM, Oh W. Rf watermark backward compatibility tests for the atsc terrestrial dtv receivers. *IEEE Transactions on Broadcasting* June 2011; **57**(2):246–252, doi:10.1109/TBC.2011.2104810.
98. Hickman A. Hbbtv testing – an approach to testing tv receiver middleware based on web standards. Https://www.w3.org/2013/10/tv-workshop/papers/webtv4_submission_10.pdf.
99. Flores-Guridi P, Guimerans G, Garella JP, Baliosian J, Grampín E, Sotelo R, Simon M. Development of a digital tv receivers test suite. *DYNA* October 2015; **82**(193):127–136, doi:10.15446/dyna.v82n193.46904.