H.264/AVC bitstream transmission simulator
User manual
(Version 0.2, December 2008)

Matteo Naccari

Introduction

This brief manual describes the software tool "`transmitter-simulator-avc`" which simulates the transmission of an H.264/AVC bitstream through an error prone channel.
When you download the zip folder of the software tool you will find inside the following items:

- *transmitter-simulator-avc.sln:* the MS Visual Studio Dot Net 2003 solution related to the source code.
- *transmitter-simulator-avc.vcxproj:* the MS Visual Studio 2019 project related to the source code.
- *error_plr_x (x ∈{3,5,10,20})*: three examples of error pattern files corresponding to Packet Loss Rates (PLRs) of 3, 5, 10, 20 %. For further details over the generation of the aforementioned files see "**S. Wenger, "Error patterns for Internet experiments", JVT-Q15-I-16r1, October 1999**".
- *config_file.txt:* an example of configuration file containing the parameters of the "`transmitter-simulator-avc`" software tool.
- *transmitter-simulator-avc_user_manual.pdf:* the pdf of this manual.
- The source code files of the "`transmitter-simulator-avc`" software tool. These are:
  - packet.h
  - parameters.h
  - simulator.h
  - packet.cpp
  - parameters.cpp
  - simulator.cpp
  - main.cpp

The source code is written in C++ using the MS Visual Studio 2019 IDE. For any bug or information you can contact me at the following email addresses:

- matteo.naccari@polimi.it
- matteo.naccari@gmail.com
- matteo.naccari@lx.it.pt

Features and functioning

The "`transmitter-simulator-avc`" software simulates the transmission of H.264/AVC bitstreams over error prone channels. The three main features of the "`transmitter-simulator-avc`" software are:

- Possibility to deal with bitstreams packetised with both the Real-time Transfer Protocol (RTP) and AnnexB standards.
- Different bitstream corruption modalities: all the packets, all the packets but the ones containing the intra coded slice and only packet containing intra coded slices.
- Possibility of generating different channel realisations starting by a unique error pattern file.

The "transmitter-simulator-avc" software simulates the transmission of H.264/AVC coded bitstreams by discarding coded packets according to a given error pattern file. The error pattern file is a file containing a sequence of byte characters ‘0’ and ‘1’ whereby ‘0’ means no transmission error and ‘1’ means transmission error. A snippet of one error pattern file is reported in Figure 1:

```
000000100000000011100100000000000101000001010000000100000000000001100
```

Figure 1: an example of error pattern contained in the error pattern file.

In order to simulate noisy transmission, the “`transmitter-simulator-avc`” software needs the parameters listed in **Error! Reference source not found.**:

Table 1: “`transmitter-simulator-avc`” input parameters description.

Parameter name	Description
Input bitstream	The H.264/AVC bitstream being corrupted
Output bitstream	The H.264/AVC corrupted bitstream. This file represents the bitstream that would be transmitted to the receiver if the channel drops coded packet according to the error pattern file
Error pattern file	The file containing the channel errors
Packet type	The standard used for the bitstream packetisation (RTP or AnnexB)
Offset	This is an unsigned integer number which represents the starting point where to read the error pattern file. By means of the offset parameters, the “`transmitter-simulator-avc`” software can simulate different channel realisations starting by the same error pattern file
Modality of corruption	The modality to corrupt the H.264/AVC bitstream specified by the “Input bitstream” parameter. The allowed modalities are: • 0: corrupts all the packets according to the error pattern file • 1: corrupts all the coded packets but the ones containing intra coded slices • 2: corrupts only packets containing intra coded slices

The bitstream transmission simulation is performed according to the following pseudo-code:

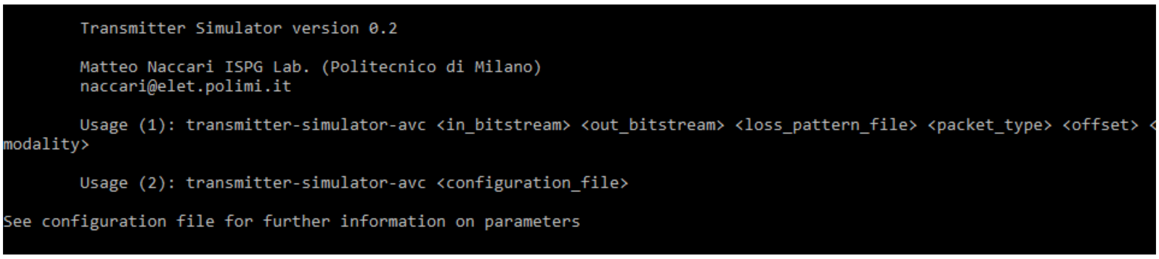
```
Open the input bitstream
Open the output bitstream
Open the error pattern file and load it into array A
set j to Offset
set mode to modality
while there are packets to read from the input bitstream do
    read the next packet from the input bitstream and put into P
    if P contains an intra coded slice AND mode == 1 then
        set writeable to 1
    end if
    if P does not contains an intra coded slice AND mode == 2 then
        set writeable to 1
    end if
    if the packet belongs to the first sequence frame OR A[j] == ‘0’ then
        write P to the output bitstream
        j++
    else if A[j] == ‘1 then
        if writeable == 1 then
            write P to the output bitstream
        else
            j++
        end if
    end if
    if j >= length(A) then
        j = 0
    end if
end do
```

Installation

The “`transmitter-simulator-avc`” source code has been developed and tested with the MS Visual Studio 2019. A CMakeList file is also provided to build the software with a different toolchain (e.g. g++ under Linux). The CMakeLists file has been automatically generated from the MS Visual Studio 2019 solution file and tested with g++ 7.5 under the `wsl` environment.

Usage

The “`transmitter-simulator-avc`” software allows two types of usages: one whereby the input parameters are passed through configuration file and the other one whereby the input parameters are passed through command line.
By typing `transmitter-simulator-avc.exe` on the MS-DOS command prompt or the Linux command shell, the software will show a little online help about its usage (see Figure 2).



```
        Transmitter Simulator version 0.2

        Matteo Naccari ISPG Lab. (Politecnico di Milano)
        naccari@elet.polimi.it

        Usage (1): transmitter-simulator-avc <in_bitstream> <out_bitstream> <loss_pattern_file> <packet_type> <offset> <
modality>

        Usage (2): transmitter-simulator-avc <configuration_file>

See configuration file for further information on parameters
```

Figure 2: output provided by the "`transmitter-simulator-avc`" software by typing `transmitter-simulator-avc.exe`.

Notice

Although all the described functionalities are implemented for both the RTP and AnnexB packetisations, extensive testing has been performed only on H.264/AVC bitstream packetised according to RTP. Therefore, it is not guaranteed the perfect functioning also for H.264/AVC bitstream packetised according to AnnexB.

Error pattern files with different format will lead to an unpredictable behaviour of the "`transmitter-simulator-avc`" software.

In order to let the "`transmitter-simulator-avc`" software to function properly, the order of the input parameters must be as the one specified in **Error! Reference source not found.**.