Performance of SoftCast and H.265 in Software Radio Video Multicasting Systems

Giuseppe Baruffa, Fabrizio Frescura
Department of Engineering
University of Perugia
Perugia, Italy
giuseppe.baruffa@unipg.it, fabrizio.frescura@unipg.it

Abstract—Wireless video distribution to multiple users suffers from the heterogeneity problem, which often results in sacrificing the video quality in order to cover a larger geographic area. The SoftCast system devised by Jakubczak and Katabi allows, in such scenarios, the delivery of a video stream with embedded graceful degradation capabilities. With the current emergence of Software Defined Radio (SDR) systems, video communications can take profit of physical layer waveform flexibility added to the ease of reprogrammability, even at the higher layer levels. In this paper, we compare the performance of SoftCast with the latest H.265 video encoding standard, in order to show its advantages. Moreover, we also present the results of an implementation of the SoftCast system using SDR devices.

Keywords—Wireless video; video multicasting; H.264; H.265; SoftCast; software defined radio

I. INTRODUCTION

Wireless video multicasting plays an important role in modern point-to-multipoint communication systems. The coverage of a large geographic area where several concurrent and heterogeneous users coexist, each one characterized by a different reception quality, is problematic when a video source with a single embedded quality layer is available [1]. Thus, layered video encoding strategies have been devised, but they are often characterized by additional complexity and by the problem of optimal layer splitting. Another solution is that of encoding a single video layer for the optimal reception from the largest possible number of users [2], or for the optimal reception by the user with the worst channel. However, in this case, the capabilities of the users characterized by the best channels are clipped to a low level. Finding an optimal trade-off between the source coding bits and the channel coding bits has been the subject of several works in the literature [3], [4], although it often requires a return link to communicate back the Channel Status Information (CSI) needed for the optimal joint source-channel coding policy. By considering the advances in the licensing of unused spectrum by Software Defined Radio (SDR) and Cognitive Radio (CR) technologies, standards such as IEEE 802.22 [5] have emerged and application scenarios like Smart Grids [6] and Internet of Things [7] have become feasible, among others.

The flexibility of CR networks and the need for high spectral efficiency offer room to adapt the generation of the video payload to the distribution channel characteristics [8] and to unintentional or hostile interference [9]. A judicious use of either the available power [10] or bitrate [11] can result in video quality performance optimization without increasing the occupied channel bandwidth or transmission power. The recently introduced SoftCast transmission scheme [12] represents a novel joint source encoding and modulation strategy that allows for an efficient use of the communication channel, and it also allows heterogeneous users to take optimal profit of their channel quality, with a single source encoding strategy and transmission applied to everyone. In this paper, we extend the SoftCast codec by considering also color information, and compare its performance with state-of-the-art video codecs such as H.264 and H.265. In all cases, we use an Orthogonal Frequency Division Multiplexing (OFDM) based subsystem for delivering the data over the RF channel. We also complement our simulations with real transmissions of the SoftCast system using commercially available SDR transmitters and receivers.

II. SYSTEM MODEL

The system can be modeled by two different sub-systems, one dealing with the source coding of the video input, and one managing the physical layer transmission, as shown in Fig. 1: the source coding subsystem will be described in detail in Sec. II.A, whereas the physical layer subsystem will be presented in...
Sec. II.B. Some modifications will be required in the physical layer when H.264/H.265 compression replaces SoftCast, as it will be detailed in Sec. III.

A. Source Coding Subsystem

Source coding follows the guidelines of the SoftCast compression algorithm [12]: it adopts a lossy compression technique based on combined spatio-temporal Discrete Cosine Transform (3D-DCT), energy-based coefficients pruning, and Hadamard Transform (HT) for energy dispersal. With reference to the top layer of Fig. 1, the source coding algorithm is described in the following. At first, the video sequence is segmented into Groups of Pictures (GoPs) of $N_{gop}$ frames. For the generic GoP pixels $v[n_1, n_2, n_3]$, $n_1$ and $n_2$ denote the spatial dimensions ($W_p \times H_p$, picture width and height), and $n_3$ denotes the temporal dimension ($0 \leq n_3 < N_{gop}$). The 3D-DCT is applied to obtain the transformed coefficient set, $V[k_1, k_2, k_3]$, using the separability property, as

$$V[k_1, k_2, k_3] = \text{DCT}_3 \{ \text{DCT}_2 \{ \text{DCT}_1 \{ v[n_1, n_2, n_3] \} \} \} , \quad (1)$$

where $\text{DCT}_n \{ \cdot \}$ is the 1D-DCT operator working along dimension $d$, and $k_1$, $k_2$, $k_3$ are the coefficient indexes in the transformed domain. Lossy compression is achieved by selecting the transform coefficients that minimize the reconstruction error at the receiver and maximize the rate-distortion (R-D) performance. However, since a mapping of the selected coefficients must be available for reconstruction, and this information size can be quite large, SoftCast operates on rectangular blocks or chunks of coefficients, by selecting all or none of the chunk coefficients at one time. Without loss of generality, we suppose that the picture can be segmented in chunks of constant size: a chunk of size $W_c \times H_c$ is defined by

$$c_{i_1,i_2,i_3}[k_1, k_2] = V[W_{i_1} + k_1, H_{i_2} + k_2, i_3],$$

$$0 \leq k_1 < W_c, \quad 0 \leq k_2 < H_c, \quad 0 \leq i_3 < N_{gop}, \quad 0 \leq i_1 < W_p,$$  

$$0 \leq i_2 < H_p, \quad 0 \leq i_3 < Hc, \quad 0 \leq i_1 < N_{gop}, \quad 0 \leq i_2 < H_c,$$  

where $i_1$, $i_2$, and $i_3$ locate the chunk on the chunks grid in the transformed domain, and the chunk energy is $\mathcal{E}_{i_1,i_2,i_3} = \sum k_1 \sum k_2 c_{i_1,i_2,i_3}[k_1, k_2]$. The energies of all the chunks are then sorted in descending order, and only the chunks corresponding to the first $N_k$ highest energies are kept for further processing, while the remaining chunks are discarded (to minimize the distortion). The number of kept chunks $N_k$ (or the fraction of kept coefficients $R_k$) linearly determines the signal occupied bandwidth. The kept chunks $s_i[k_1, k_2], 0 \leq i < N_k$, are supplemented by the selection mapping $s_i = M(i;d), d = 1, 2, 3$, which encodes the original position of the $i$-th kept chunk, such as

$$s_i[k_1, k_2] = c_{M(i;1),M(i;2),M(i;3)}[k_1, k_2] . \quad (3)$$

SoftCast randomizes and linearly encodes the chunk coefficients by applying the HT along slices covering different chunks. In this way, more energetic chunks and less energetic ones are fused together, reducing the dynamic range of the generated signal, after that their average value has been removed. The randomized signal is given by

$$S_p[m] = \text{HT} \left\{ I_p \left( g_i \left( s_i[k_1, k_2] - \mu_i \right) \right) \right\}, \quad (4)$$

where $p$ is the slice index, $m$ is the slice sample index, $g_i$ is an amplitude scaling factor, $I_p(\cdot)$ is an interleaving operator that selects a slice of the chunk coefficients of proper size, $\text{HT}(\cdot)$ is the 1D-HT operator of proper size, and $\mu_i = \sum k_1 \sum k_2 s_i[k_1, k_2]$ is the chunk average value. The scaling factor $g_i$ has been calculated according to the inverse quartic law of the chunk variance, as recommended in [12]. After this operation, the number of samples in all the slices is the same as the number of coefficients in all the kept chunks, i.e., $H_p W_p N_c$ samples. The size of the HT can be chosen to maximize the contribution of different energy chunks into one slice, and it can be different for some slices in order to respect the final sample size bounds and the admitted HT sizes. Clearly, in case of color video transmission, some of the above steps should be repeated in parallel, while the remaining steps follow serially. For a 4:2:0 YUV chroma-subsampled sequence, the 3D-DCT is repeated independently on the three color components, but chunk selection is based on the Y component only, and the selected chunks of the U and V components are spatially located in the same positions of the selected Y chunks (proper rescaling of U and V chunks position and size is performed). After chunk selection, the chunks from the color components are assembled together, and then they share a common encoding and modulation path.

B. Physical Layer Subsystem

With reference to the bottom layer of Fig. 1, the HT transformed samples are mapped into complex data symbols as $C_D[q] = S[2q] + j S[2q + 1]$, where $S[m]$ is a concatenation, for all $p$, of the slices $S_p[m]$. The additional metadata required to recover the original video signal are transmitted in a conventional way, by means of Bose-Chaudhuri-Hocquenghem (BCH) channel coding and differential phase shift keying (DPSK) modulation. In particular, information such as frame width and height $W_p \times H_p$, chunk width and height $W_c \times H_c$, and number of chunks $N_k$ is protected by a more robust BCH(255, 71) code, whereas the chunk mapping $M(i;d)$, the scaling factors $g_i$ (after quantization with $N_q$ bits), and the mean values $\mu_i$ (after quantization with $N_q$ bits) are protected by a less robust BCH(127, 71) code. The physical layer adopts OFDM, characterized by $N_{tot}$ total carriers, $N_{act}$ active carriers, and a cyclic prefix of $N_c$ samples. Each OFDM block is built, in the frequency domain, by a $\text{framer}$, which juxtaposes and combines the source coding data symbols $C_D[q]$ with pilot symbols $C_P[q]$ (used for synchronization...
and channel estimation purposes) and metadata symbols
\(C_u[n]\): \(X_s[k]\) represents the symbols available at the framer
output for the \(b\)-th OFDM block in the GoP (symbols in the
last block may contain random data used to adjust the
transmission rate, so as to have an integer number of OFDM
blocks in one GoP). The samples of the \(b\)-th OFDM block in the
GoP are expressed by

\[
x_b[n] = \begin{cases} 
\sum_{k=0}^{N_c-1} X_s[k] e^{j \frac{2\pi}{N_{\text{TOT}}} (N_{\Delta} \cdot 2^{-1})}, & -N_{\Delta} \leq n < N_{\text{TOT}}, \\
0, & \text{else}, \end{cases}
\]

and the overall signal representing one GoP is

\[
x[n] = \sum_{b=0}^{N_{b}-1} x[n - bN_c],
\]

where \(N_c = N_{\text{TOT}} + N_{\Delta}\) is the OFDM block length, and \(N_b\) is
the number of OFDM blocks in one GoP.

### III. Simulated System Performance

The system described in Sec. II has been implemented
using MATLAB and other tools. The physical layer and source
coding subsystems are implemented exclusively in MATLAB.
In order to perform a comparison with other compression
standards, in the source coding subsystem we have used
FFmpeg [13] as video coding tool for the H.264 and H.265
video compression standards [14], while keeping essentially
the same OFDM strategy in these cases. The main differences
between the H.26x and the SoftCast physical layer subsystems are:

- concatenated channel coding protects the H.26x
  bitstream, with a Reed-Solomon RS(204, 188) outer
code and a punctured convolutional inner code
(puncturing rates of 1/2, 2/3, 3/4, 5/6, and 7/8);
- rectangular quadrature amplitude modulation (QAM)
  with \(M = 4, 16, \) or 64 constellation points maps the
  protected data on the frame symbols;
- data interleaving takes place between the R-S encoder
  and the convolutional encoder, as well as before
  mapping the channel encoded bits on the data carriers.

The complete system has been tested by means of Monte
Carlo simulations, to assess either the R-D performance or the
PSNR vs. Carrier-to-Noise ratio (C/N) performance. A 512-
frame video composed by the first 32 frames of 16 clips (akiyo,
bus, coastguard, crew, flower, football, foreman, harbor,
husky, ice, news, soccer, stefan, tempete, tennis, and waterfall)
with CIF resolution (352×288, 30 frames/s) and YUV 4:2:0
format [15], has been adopted [12]. Only the luminance (Y)
component of the video has been used to evaluate the
distortion, in terms of PSNR.

### A. R-D Performance

The source coding rate of the systems under test has been
measured in terms of occupied bandwidth, since SoftCast uses
transformed coefficient samples and H.26x uses binary source
samples. In all cases, the bandwidth \(B_V\) has been measured as

\[
B_V = \frac{N_{\text{ACT}}}{N_{\text{TOT}}} \frac{f_s}{N_{\text{TOT}}} \frac{N_{C}}{N_{\text{GOP}}} R_V,
\]

where \(f_s\) is the sample rate produced by the complete system,
\(N_{\Delta}\) is the effective number of OFDM blocks per GoP, and \(R_V\)
is the video frame rate. It should be noted that, in the SoftCast
case, the sample rate is fixed as there is always (by design) a
fixed number of blocks in a GoP. On the other side, in the
H.26x case, we may use either a constant quality compression
policy specified by the constant rate factor (CRF), which
produces a variable number of bits per GoP, or a constant
bitrate compression policy (constant bit rate, CBR), which
produces an approximately constant number of bits per GoP. In
the former case, the occupied bandwidth has been calculated on
the best (\(N_{\text{ACT}} = \min_{\text{GoP}} N_{\Delta}\)), average (\(N_{\text{ACT}} = \text{avg}_{\text{GoP}} N_{\Delta}\)), or
worst case (\(N_{\text{ACT}} = \max_{\text{GoP}} N_{\Delta}\)). We have first tested how the
size of the chunks influences the SoftCast encoding
performance. Fig. 2 shows the obtained PSNR for three
different chunk sizes: 22×18, 44×36, and 88×72. The resulting
distortion curves suggest that larger chunk sizes are to be
preferred. In our following discussion and experiments, thus,
we have used a chunk dimension of 44×36, since the 88×72
case provides almost no benefit. Fig. 3 shows the rate-
distortion performance comparison of H.264, H.265, and
SoftCast based systems. In all cases the GoP is fixed to a size
of 8 frames. For H.26x, we have plotted the minimum, average,
and maximum bandwidths, using 16-QAM and 3/4 code rate.

![Fig. 2. R-D performance of SoftCast for various chunk sizes.](image-url)
Fig. 3. R-D performance of the SoftCast, H.264, and H.265 systems.

If performance is compared in terms of average bandwidth, SoftCast has an advantage over H.26x only when there is no scarcity of available bandwidth (i.e., for $B_p > 2.5$ MHz). However, if we consider the maximum bandwidth, then the advantage of H.26x over SoftCast is less pronounced, valid only when $B_p > 1.5$ MHz. The performance curves of H.265 confirm the well-known advantage over H.264, with a bandwidth gain of about 0.2 MHz for the same PSNR quality.

B. PSNR vs. C/N Performance

Several types of impairments have been added to the system, in order to simulate the quality of the signal at the input of a real receiver. Namely:

- quantization and clipping of the signal, both at the transmitter and receiver sides, models digital-to-analog (DAC) and analog-to-digital conversion (ADC);
- additive white Gaussian noise (AWGN) models the receiver thermal noise and defines the C/N ratio;
- line-of-sight (LOS) and non-LOS (NLOS) channels model the transmission over a realistic static multipath channel (according to models well-known in the literature), consider the global path loss, and delay the signal by a fractional sample interval;
- carrier frequency offset (CFO) models the effect of transmitter and receiver oscillators instability;
- DC offset (DCO) models incorrect isolation of the local oscillator (LO) carrier from the useful signal.

The received signal can be written as

$$y[n] = \frac{1}{k_{RX}} Q_{ADC} \left[ \left( \frac{k_{RX}}{k_{TX}} Q_{ADC} \left( k_{TX} x[n] \right) \ast h_{CH}[n] + D_{DC} \right) e^{j\Delta \omega n} + w[n] \right],$$

where $Q_{ADC}\{r\}$ is the quantization/clipping operator defined by

$$Q_{ADC}\{r\} = \begin{cases} +1 & , r \geq 1 \\ 2^{\frac{Q-1}{2}}, & -1 < r < 1 \\ -1 & , r \leq -1 \end{cases},$$

$B_{DAC}$ and $B_{ADC}$ are the number of quantization bits used by the DAC and ADC, respectively, $k_{TX}$ and $k_{RX}$ are normalization constants used to control the whole gain of the system, $h_{CH}[n]$ is the (N)LOS channel impulse response, $D_{DC}$ is the DCO value, $\Delta_\omega$ is the CFO due to TX and RX oscillators, and $w[n]$ represents the AWGN term. At the receiving side, synchronization is performed both at the OFDM block and GoP levels. A well-known time and frequency synchronization algorithm [16] has been adopted to estimate the beginning of the OFDM blocks and the CFO, whereas the DCO can be estimated by finding the mean of the received signal. In the following, we have assumed CFO and DCO to be recovered and corrected. Moreover, DAC and ADC behavior is simulated according to the USRP device specifications (best case), with $B_{DAC} = 16$ bits and $B_{ADC} = 14$ bits. We have used $N_{carrier} = 2048$ carriers, of which only $N_{carrier} = 1705$ are active, and a cyclic prefix length of 1/4 of the useful part length. Pilot carriers have been inserted using a fixed-position pattern (with an average inter-pilot distance of 38 carriers), to be used also for frequency synchronization, and a scattered-position pattern (average pilot distance of 12 carriers). The channel transfer function is estimated at the receiver by bilinear interpolation on the time-frequency grid, using the pilot carriers as reference points, followed by bilinear low-pass filtering for noise smoothing. Since from the results discussed in Sec. III-A there is a clear advantage in using H.265 over H.264, we only performed simulations using H.265. The default FFmpeg settings for error resilience and concealment in H.265 have been used. Fig. 4 presents some of the obtained results. In the following, we consider that there is a fixed channel bandwidth available, and each tested system must comply to that value. Thus, for SoftCast (SC), the legend reports the coefficient ratio $R_e$ and the effective bandwidth $B_p$, which is constant. On the other side, for H.265, the CRF and the maximum bandwidth are shown. The chosen bandwidth values are 0.5 MHz, 1 MHz, 1.5 MHz, 2 MHz, 2.5 MHz, and 3 MHz. By observing the performance curves obtained with a comparable signal bandwidth, we notice that SoftCast (solid lines) can achieve a C/N advantage over H.265 (dotted lines) of approximately 2 dB, when H.265 is encoded with rate 1/2 and modulated with 16-QAM (Fig. 4a). In Fig. 4b, H.265 is modulated with 16-QAM and encoded with rate 3/4: in this case, the advantage at low values of C/N for SoftCast is of about 6 dB. This means that users that cannot correctly decode the H.265 stream, due to poor C/N conditions, are still able to receive and decode the SoftCast stream with a sufficient quality for visual intelligibility of the video contents.
Moreover, for higher values of C/N, the PSNR of H.265 is limited by the given bandwidth, whereas the PSNR of SoftCast improves with the C/N. Fig. 5 shows the same type of performance comparison when the simulation is carried out on a multipath channel, instead of the AWGN one. A static hilly terrain (HT) channel, characterized by 12 taps in the impulse response, has been generated according to COST 207 specifications [17], for the maximum expected sample rate produced by both systems. Then, for each one of the curves reported in the figure, the channel impulse response has been resampled to the effectively adopted sample rate, and the transmission and reception chain simulated. The results confirm the superiority of SoftCast over H.26x at low values of C/N: H.26x provides valuable video quality only starting from about 21-22 dB of C/N.

IV. EXPERIMENTAL RESULTS

Off-line, nonreal-time experimental transmission and reception tests of SoftCast, under real RF channel conditions, have been carried out using SDR hardware and GNU radio [18]. For our experiments, we have used either the Ettus USRP N210 [19] or the HackRF One [20] as transmission devices, and either the HackRF One or a Realtek chip-based USB dongle (RTL-SDR) [21] as reception devices. A quantized version of the generated base-band signal has been saved as a binary file in the complex interleaved 16-bit format, using MATLAB. Then, the file has been used in a GNU radio flowgraph as input source to a USRP device sink block. A dual flowgraph has been used at the receiving side, with a binary file sink block cascaded to an RTL-SDR source block. The received binary file is then used in MATLAB as digital source to perform signal reception and video decoding. In this experiment, the signal is propagated exclusively inside of our laboratory, with antennas in line of sight. The USRP device was selected for transmission using a log-periodic antenna, and the signal has been received by the RTL-SDR device with a stylus antenna. Several tests were performed at different values of C/N, namely 5.5 dB, 10.7 dB, 16.2 dB, 18.4 dB, and 27.5 dB: C/N has been varied by controlling the USRP transmission gain. In the tested cases, the chosen compression ratio $R_c = 0.348$ corresponds to an effective occupied bandwidth $B_o = 0.835$ MHz (for both devices, a sampling rate of $f_s = 1$ Msample/s was used). In all cases, the video sequence has been correctly received and decoded with a quality proportional to the C/N; these results agree with the expected performance, as obtained by the simulations. Fig. 6 shows some results for the last case, i.e., C/N=27.5 dB. Since the transmitting and receiving antennas were in LOS, the received spectrum (Fig. 6a) is almost flat and experiences only mild multipath effects, and the received video quality is good (Fig. 6b). The received metadata constellation points, after channel estimation and zero-forcing equalization, are concentrated around the nominal constellation points of the binary DPSK modulation, as shown in Fig. 6c. As a term of comparison, Fig. 6d displays the received values of the data symbols, after channel estimation.
and equalization as well. Finally, Fig. 6e shows the estimated CFO, of about 135 Hz, before being compensated by digital rotation at the receiver: in this case, the CFO is generated by the TX and RX oscillators frequency instability.

V. CONCLUSION

In this paper, we have compared the performance of the SoftCast video compression algorithm to that of the H.264 and H.265 video codecs, in terms of both rate-distortion performance and PSNR vs. C/N performance on AWGN and multipath channels. The results have shown the advantage of SoftCast over H.265 at low values of C/N, electing SoftCast as the best choice for all the cases when multicast video quality must degrade gracefully in concordance with users' heterogeneity. We have also presented the results of a nonrealtime transmission experiment of the SoftCast system, implemented on SDR devices such as the USRP N210 and an RTL-SDR dongle.

![Image](image1.png)

Fig. 6. Received signal: power spectrum (a), video frame sample (b), metadata symbols (c), data symbols (d), and estimated CFO (e).

ACKNOWLEDGMENT

The authors would like to thank Lorenzo Germani for his help in setting up the simulations and performing the experiments.

REFERENCES


