Application Layer Source-Channel Video Coding for Transmission with Smartphones over Satellite Channels

Igor Bisio, Aldo Grattarola,Fabio Lavagetto, Giulio Luzzati, Mario Marchese
DITEN - University of Genoa
Genoa, Italy
Email: igor.bisio, aldo.grattarola , fabio.lavagetto, giulio.luzzati, mario.marchese@unige.it

***Abstract*—Transmitting video with portable devices (such as smartphone), over heterogeneous networks with satellite portions, to a Remote Monitoring Host (RMH) represents a hot scientific and technical topic. Unfortunately, heterogeneity often implies impairments such as packet losses, due to errors and congestion, which negatively affect the video quality. To guarantee a satisfactory video fruition at the RMH, an application layer joint coding algorithm for video transmission is presented. It adaptively applies both video compression and encoding to protect the sent video stream, at the application layer, on the basis of the overall network condition estimated in terms of both maximum allowable throughput of the network and quality (packet cancellations or *lossiness*). A preliminarily performance study, carried out with real implementation of the algorithm, compares the joint coding against fixed schemes.**

Keywords*—*Application layer; joint source channel coding; disaster recovery.

I. Introduction

The recent spreading of mobile computing platform such as those constituted by smartphones has provided grounds for ubiquitous connectivity. Much research has been done in the field, especially in the light of the recent advances in wireless communication [1], and [2]. The reference network considered in this paper is composed of heterogeneous portions, which include two main components: radio and satellite, which allows wide coverages. Offering real services with a specific guarantee of Quality of Service (QoS) over these hybrid systems is very challenging, as implies the solution of research and development issues due to the peculiarity of the application environment: efficient communication networks composed of mobile and fixed nodes operating with radio and satellite links. For this reason, the study and the tests in this paper is concentrated on the integrated wireless-satellite components of the network, where also much work has been done [3], [4]. These portions, unlike wired channels where information loss is mainly imputable to network or resource congestion, wireless media are characterized by low Signal-to-Noise Ratios (SNRs). This leads to Bit Error Rates (BERs), which amount may range from negligible to almost completely impairing. Moreover, when dealing with wireless channel the time invariance assumption can no longer hold: typical wireless channels may exhibit extremely quick dynamics, due to a number of factors, such as multipath fading, shadowing, radio interferences and weather conditions.

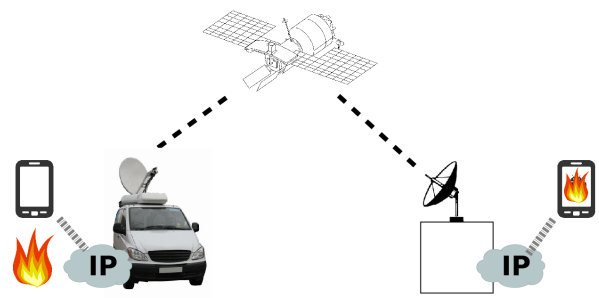

Fig. 1: The reference scenario

The specific scenario considered in this paper (reported in Fig.1) consider the case in which there is the need for transmitting a video stream over a satellite link. It often happens when an emergency link need to be established between an incident site and a RMH where a command center is operating. Unfortunately, heterogeneity often implies impairments such as packet losses, due to errors and congestion, which negatively affect the video quality. With this kind of scenario in mind, a static management of how the information is compressed and protected throughout its flow clearly constitutes a non optimal choice: one would want to be able to dynamically adapt the information flow by opportunely tuning the amount of information offered to the transmitting apparatus and how it is being protected, possibly jointly considering the impact each of these these tunings has on each other and on the whole system performance.

In more detail, to guarantee a satisfactory video fruition at the RMH, an application layer joint coding algorithm for video transmission is presented. It adaptively applies both video compression and encoding to protect the sent video stream, at the application layer, on the basis of the overall network condition estimated in terms of both maximum allowable throughput of the network and quality (packet cancellations or *lossiness*). By choosing to exploit the tools provided at the application layer, we develop an application layer joint coder completely oblivious of what happens at the lower network

 ISBN: 978-1-61208-317-9

layers, thus being able to be deployed on top of existing infrastructure without any need for rewiring or replacing hardware components. Application layer entities talk to each other via datagrams, so whenever one packet is sent, it is either delivered (possibly with some amount of delay) or lost as a whole. This can be modelled by means of a Packet Erasure Channel (PEC) model, and it is the reason we chose to protect information using a packet level channel code as detailed in the next Sections.

The remainder of the paper is organized as follows: Section II surveys the state of the art regarding the application layer coding and source-channel join coding problems; in Section III is described the proposed Application Layer Joint Coding (ALJC) approach. The simulative scenario developed to validate the proposed algorithm and the obtained numerical results are discussed in Section IV. The obtained results represent a initial study of the performance whose development has been proposed in [5]. Finally, the conclusions are drawn in Section V.

II. State of Art

In the literature, in particular in the field of satellite and space communications, it is argued that application-layer coding, obtained by applying redundancy at the application layer to protect the sent information, may be used to efficiently recover original data by guaranteeing flexibility and easy-configurability [6]. Nevertheless, the advantages of applying coding strategies at the application layer may improve the performance only in systems with low channel error rates because high error rates imply high levels of redundancy thus causing information losses due to congestion over a network. In more detail, starting from the Separation Theorem formalized by Shannon in [7] which is based on some assumptions such as infinite coding length, no constraints on complexity and on delay, in [8] the existence of two performance regions has been formally demonstrated. In a region, the employment of application layer coding is significantly advantageous while it is quite useless in the second. The first performance region is represented by systems that experience low level of intrinsic loss probability. On the contrary, applying such coding in high intrinsic loss probability systems is not advantageous because the high level of necessary redundancy causes a congestion growing. Differently, [9] proposed a slightly different point of view of the problem, whereas increasing protection does not directly result in increased offered load: in this sense, an end-to-end distortion minimization algorithm is devised. However, to the best of the authors' knowledge, literature lacks investigation upon actual implementation of a joint source-channel coding at the application layer, whereas proposed approaches take channel state knowledge for granted.

The aforementioned results inspired, as done similarly in [10], the rationale under this paper is that the boundary between the two regions can be moved so increasing the region size where the usage of the application layer coding is useful. In technical words, if the information sent from the sources is previously compressed the load offered to the network will be reduced. As a consequence, the redundancy employment does not cause a critical network load increasing.

For this reason, when dealing with practical radio networks it may be beneficial to handle compression and protection jointly, taking into account how one affects the other also. Moreover, differently from the aforementioned approach, our work introduces a method to estimate of the overall network condition estimated in terms of both maximum allowable throughput of the network and quality (packet cancellations or *lossiness*). It allows adapting, dynamically, compression and protection applied, in the case of this paper, to the transmitted video.

III. The Heuristic ALJC Approach

The technical reference of the proposed ALJC approach is reported in Fig.1. In the depicted scenario, video streams are transmitted, with portable devices (such as smartphone), over terrestrial/satellite networks, to a RMH. It may support emergency and rescue operations after crisis situations. For this reason, the practical implementation for the proposed ALJC has been developed on the *Android* operating system, by building two separate applications: a transmitter and a receiver. The chosen source encoding for the video frames is MJPEG due to its implementation and management easiness, as well as its intrinsic resiliency: it does not remove temporal correlation, so that losses and errors do not propagate in consequent frames, i.e. the stream does not lose synchronization even if the link temporarily drops. For what concerns the channel coding, LDPC [11] has been devised as a practical and computationally feasible solution. Finally, the stream of information is then handled to the UDP transport protocol.

Working on *Android* operating systems, we already have control of the source coder, by means of a JPEG encoding class which allows to specify a compression index q, in the $[0, 100]$ range, from worst to best quality, respectively. Considering the aforementioned optimization framework $s_{1k} = q$, $\forall k \in [1, K]$, $M = 1$. The chosen resolution for the video frames is the standardized resolution QCIF (176 x 144 pixels) .

The LDPC codec has been ported from an existing implementation [12]. The source code has been adapted and wrapped to be used as a native library within the *Android* Native Development Kit (NDK). It is a packet level LDPC code, meaning that it allows recovering of whole packets of data (i.e., a codeword is a set of packets). The encoder is used to dynamically set the employed code rate fec (i.e., the number of packets containing information over the overall number of packets composing a codeword). Considering again the aforementioned optimization framework $c_{1k} = fec$, $\forall k \in [1, K]$, $N = 1$.

The tests presented in this paper were carried out using packets 1024 bits long, which compose codewords 35 packets long. However, the LDPC mechanism needs some additional control information to be transmitted along with the source information flow, namely sequence numbers which identify codewords, packets within codewords and contents within the packets. This information is carried by a 24 bits additional header, leading what we call an *application layer packet* to be 1048 bits long.

The adaptiveness is made possible by means of a feedback (return) channel, that is, the receiver periodically sends information to the transmitter in the form of cumulative *ack* packets that carry information about how many packets have been lost, sent each time a new codeword is received, which allows the transmitter to make estimates about the overall network state. The return channel in our experimentation is error-free. In this sense, the key point for a system that aims at managing the source and channel coding jointly is to estimate how fast the hybrid terrestrial/satellite network is able to deliver the information and how vulnerable is the information in the process of traversing the entire network, in particular the radio/satellite portions. The estimation of the maximum allowable throughput of the network is made by using a particular packet buffer, which is filled by the LDPC encoder and it is emptied when an *ack* arrives (i.e., a packet is not removed after it has been sent, but it remains in the buffer until it is marked as *acked* or a *timeout* expires). The network service gross rate is given by the rate at which this buffer is being emptied.

On the other hand, since *ack* packets carry the lost packet count, provided by the LDPC decoder on the receiving side, we can use this amount as a measure of the quality (packet cancellations or *lossiness*).

The adaptive ALJC algorithm has the granularity of a codeword and, for each decision stage, it first assess the amount of protection needed to successfully traverse the entire hybrid network, then it tries to best exploit the remaining packets to try to fit in frames with the goal of maintaining a target frame rate expressed in Frame-Per-Second (FPS_{target}) and equal to, in this paper, 10 frames per second. To do this, we firstly tabulated the image size associated to each compression factor from 0 to 100 then inverted the function and computed a polynomial fitting. The mentioned actions have been performed in order to be able to obtain the compression quality needed as a function of the desired image (i.e., the single MJPEG video frame) file size.

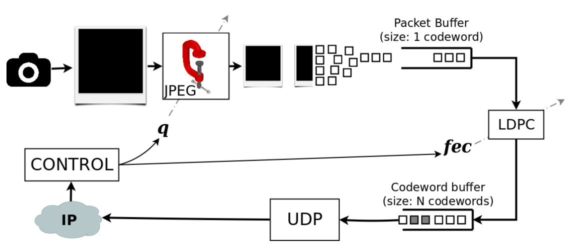

Fig. 2: The structure of the framework implementation

IV. Performance Investigation

A. Testbed

We realized a testbed to simulate the scenario depicted in the introduction. In this configuration, two separate *Android* devices communicate through a WiFi local network connected to a machine which emulates the effect of a satellite channel. On the receiving side, another WiFi network is used to interconnect the second device, which is supposed acting at the RMH.

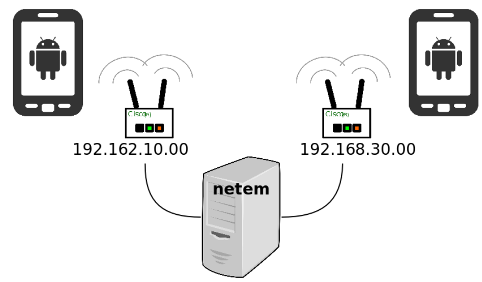

Fig. 3: The terrestrial/satellite network emulation apparatus used

This is reported in Fig.3, and it consists of a regular PC running with a Linux based operating system and equipped with two network interface cards, each connected to a WiFi access point. This way two separate sub-networks are created, and by using the *netem* tool it is possible to easily manage traffic going out of each interface, by tuning the bandwidth, the packet loss and bit error rate due to the emulated satellite channel, the delay and the packet buffer queuing policy.

B. Scenarios and Performance Metrics

TABLE I: The test scenarios

	Bandwidth	BER
A	400 Kbps	0 %
B	400 Kbps	10 %
C	180 Kbps	35 %
D	180 Kbps	0 %
E	180 Kbps	10 %
F	180 Kbps	35 %

TABLE II: The transmitter application source and channel coding settings

	FEC Rate	Compression Index
ALJC	*dynamic*	*dynamic*
Minimum Protection	30/35	60
Maximum Protection	4/35	20

In order to evaluate the behaviour of the ALJC system, we compared with two antipodal and static policies (i.e., minimal protection and maximum protection (see TABLE II). The compression index has been set to 20 in the case of maximum protection in order to provide a minimal intelligibility of the image (when equal to 0 the image information is almost completely lost), and to 60 because it conveys a frame image size that lies in the middle of the range. The test runs evaluate each coding choice behaviour during three minutes long sessions, exploiting the aforementioned network emulating machine that simulates different bandwidth and BER conditions experienced by the satellite link. A second run of tests deals with the system's adaptation capabilities in

time varying conditions.
We devised six plausible scenarios, listed and described accurately in Table I. In every scenario we had the network emulator introduce a 600 [ms] propagation delay to simulate a the presence of a geostationary satellite link in the network, as described in the introduction. In order to be able to evaluate and correctly understand how each coding policy actually behaves, we had to devise the key factors to measure in a video streaming framework. Intuitively, from a qualitative point of view it would be desirable for such a system to provide the highest possible image quality, as well as a fluent video reproduction, and naturally, a minimal information loss due to network impairments.

In order to measure the quality of individual frames of the MJPEG sequence, we utilize a well known metric, the Structural SIMilarity index. It was first introduced in [13], and it provides a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality. The SSIM represents a good choice since it follows the MOS more closely than other indexes such as the Peak Signal to Noise Ratio (PSNR) or Mean Square Error (MSE). The SSIM index is computed over small patches of image, and the whole image index is obtained by averaging the individual patches' values. The individual SSIM value for a patch is given by

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (1)$$

where x and y are the two small image blocks, μ_i is the i-th block pixel value average, σ_i is the i-th block pixel standard deviation, and

$$\sigma_{ij} = \frac{1}{M-1}\sum_{k=1}^{N}(i_k - \mu_i)(j_k - \mu_j) \quad (2)$$

where M is the number of pixel contained in the patch.

The SSIM index ranges from 0 (completely uncorrelated images) to 1 (identical images), and this is a useful feature since it can be considered a degradation factor.

In order to evaluate the performance of a given test run we devised a performance index with the following requirements

- it must reward high quality frames
- it has to reward a fluent video stream, i.e. a high frame throughput
- it must penalize corrupted or lost frames

We found that the following index S satisfies such requirements

$$S = \frac{\sum_{i=1}^{M} SSIM(f_i, \widehat{f_i}) \cdot f_{received}^{TOT}}{T_{sim}} \quad (3)$$

and can be interpreted as a *quality-weighted average framerate*.

For experimental purposes, in order to collect the information needed to compute such metrics, we arranged the implemented smartphone applications to write down each image frame: the transmitter writes a full quality version, while the receiver application records the source encoded version (i.e., including information loss due to compression). The frame written out by the smartphone applications are temporally referenced, so that it is possible to also infer information about the frame rate.

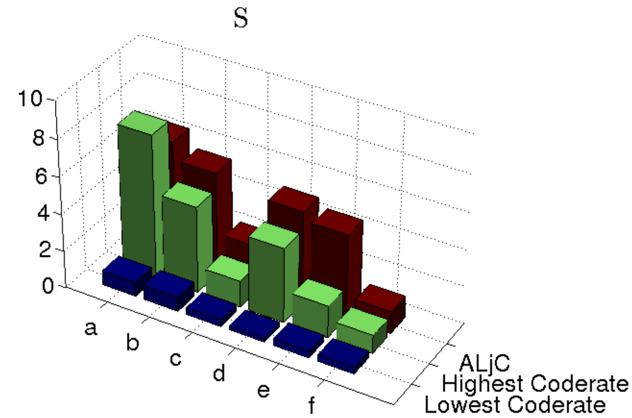

Fig. 4: Results of the simulation runs

A test run is structured in the following steps:

1) The smartphone starts streaming to the receiver, which is individuated through its IP address.
2) When the duration limit is reached, the transmitter stops broadcasting and sends to the server the reference file containing the original full quality frames.
3) As soon as it is done sending, the receiver sends its data, containing the received (and possibly corrupted) video frames.

C. Emulation Campaigns Results

In this section we show how our joint coding framework behaves in a stationary network condition (i.e., within channels whose characteristics do not vary over time), referring to Fig.4. As mentioned in Section IV-B, scenarios A, B and C, which set a bandwidth of 400 kbps and different BER conditions.

Scenario A simulates a wideband, error free channel: in this particular case the adaptive ALJC policy seems to perform slightly worse than a minimum protection one. This is reasonable, since the latter consistently employs high quality compression and waste a little quantity of bandwidth for protection, while ALJC more cautiously because the compression tuning "oscillates" due to a non-perfect network condition estimation in terms of both maximum allowable throughput of the network and quality. There is a minimal amount of lost packets to be imputable to kernel drops due to busy CPU. Naturally, the maximum protection approach here is the worst scoring, due to the unnecessarily poor quality of the frames and the low throughput due to high bandwidth waste. Scenario B is much more interesting: a significant BER (10%) causes the minimum protection policy to lose a substantial amount of packets, thus bringing down the overall sequence PSNR.
A very high BER (35%) represents a highly challenging

channel: here the performance index cannot be as high as in previous scenarios, because the ALJC policy needs to cut down on the offered load by using more aggressive compression in order to let the frames through the narrower window left by the channel encoder. At the same time, the minimal protection approach now loses nearly half the information fed into the network, while the conservative maximum protection policy is not able to transport a sufficient amount of information.

Scenarios D, E and F simulate narrowband channels instead. The behaviour is roughly the same as in the previous scenarios.

V. Conclusions

In this paper we introduced an application layer joint coding algorithm for video transmission. It allows the transmission of video streams on networks based on time varying and possibly lossy networks, such as those entailed by hybrid terrestial/satellite networks. It is shown that using only information available at the application layer, we can implement a system that outperforms oblivious static coding under nearly every network condition. From the practical viewpoint, it adaptively applies both video compression and encoding to protect the sent video stream, at the application layer, on the basis of the overall network condition estimated in terms of both maximum allowable throughput of the network and quality (packet cancellations or *lossiness*).

References

[1] G. Araniti, M. Condoluci, A. Molinaro, A. Iera, and J. Cosmas, “Low complexity subgroup formation in lte systems,” in *Broadband Multimedia Systems and Broadcasting (BMSB), 2013 IEEE International Symposium on*, 2013, pp. 1–6.

[2] L. Militano, M. Condoluci, G. Araniti, and A. Iera, “Multicast service delivery solutions in lte-advanced systems,” in *Communications (ICC), 2013 IEEE International Conference on*, 2013, pp. 5954–5958.

[3] S. Mukherjee, M. De Sanctis, T. Rossi, E. Cianca, M. Ruggieri, and R. Prasad, “On the optimization of dvb-s2 links in ehf bands,” in *Aerospace Conference, 2010 IEEE*, 2010, pp. 1–11.

[4] M. De Sanctis, E. Cianca, and M. Ruggieri, “Ip based routing algorithms for leo satellite networks in near-polar orbits,” in *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, vol. 3, 2003, pp. 3-1273–3-1280.

[5] I. Bisio, A. Grattarola, G. Luzzati, F. Lavagetto, and M. Marchese, “Performance evaluation of application layer joint coding for video transmission with smartphones over terrestrial/satellite emergency networks,” in *International Communications Conference (ICC 2014), 2014 IEEE*, 2014, submitted.

[6] T. de Cola, H. Ernst, and M. Marchese, “Performance analysis of ccsds file delivery protocol and erasure coding techniques in deep space environments,” *Comput. Netw.*, vol. 51, no. 14, pp. 4032–4049, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2007.04.015

[7] C. E. Shannon, “Coding theorems for a discrete source with a fidelity criterion,” *Institute of Radio Engineers, International Convention Record*, vol. 7 (part 4), pp. 142–163, 1959.

[8] Y. Choi and P. Momcilovic, “On effectiveness of application-layer coding,” *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6673–6691, 2011.

[9] F. Zhai, Y. Eisenberg, T. Pappas, R. Berry, and A. Katsaggelos, “An integrated joint source-channel coding framework for video transmission over packet lossy networks,” in *Image Processing, 2004. ICIP ’04. 2004 International Conference on*, vol. 4, 2004, pp. 2531–2534 Vol. 4.

[10] I. Bisio, F. Lavagetto, and M. Marchese, “Application layer joint coding for image transmission over deep space channels,” in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, 2011, pp. 1–6.

[11] R. Gallager, “Low-density parity-check codes,” *Information Theory, IRE Transactions on*, vol. 8, no. 1, pp. 21–28, 1962.

[12] “Planete-bcast, inria, ldpc codes download page,” http://planete-bcast.inrialpes.fr/article.php3?id_article=16.

[13] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.