Low-Latency Video Streaming with Congestion Control in Mobile Ad-Hoc Networks

Claudio Greco, *Member, IEEE,* Marco Cagnazzo, *Senior Member, IEEE,* and Béatrice Pesquet-Popescu, *Senior Member, IEEE*

***Abstract*— In this paper, we address the challenge of delivering a video stream, encoded with multiple descriptions, in a mobile ad-hoc environment with low-latency constraints. This kind of application is meant to provide an efficient and reliable video communication tool in scenarios where the deployment of an infrastructure is not feasible, such as military and disaster relief applications. First, we present a recently proposed protocol that employs a reliable form of one-hop broadcast to build an efficient overlay network according to a multi-objective function that minimises the number of packets injected in the network and to maximises the path diversity among descriptions. Then, we introduce the main contribution of this article: a cross-layer congestion control strategy where the MAC layer is video-coding aware and adjusts its transmission parameters (namely, the RTS retry limit) via congestion/distortion optimisation. The main challenge in this approach is providing a reliable estimation of congestion and distortion, given the limited information available at each node. Our simulations show that, if a stringent constraint of low delay is imposed, our technique grants a consistent gain in terms both of both PSNR and delay reduction, for bitrates up to a few megabits per second.**

***Index Terms*—Congestion-Distortion Model, Cooperative Systems, Cross-layer Design, Mobile Ad-Hoc Networks, Multiple Description Coding, Multimedia Communication.**

I. Introduction

A mobile ad-hoc network (MANET) is a dynamic network of mobile devices connected by wireless links, self-organised in a mesh topology [1, 2]. MANETs offer a set of properties – flexibility, ease of deployment, robustness, *etc.* – that makes them appealing in environments without preexisting infrastructure. Possible applications range from crisis management service, such as military and rescue operations (*e.g.*, to provide battlefield awareness and data dissemination), to business environments, such as video conferencing outside the office (*e.g.*, to brief clients on a given assignment), to recreational contexts (*e.g.*, to allow user to view a live stream of an event they are attending). In all these scenarios, real-time video streaming is an application of major interest [3].

The challenge of delivering real-time video streams in unreliable networks, such as a MANET, can be efficiently addressed by ensuring a partial loss resiliency with Multiple Description Coding (MDC) [4]. In MDC, the source signal (be it an image, a video, or an audio signal) is encoded in N independent streams, referred to as *descriptions*. Each description can be decoded independently from the others, and each further description improves the quality of the reconstructed signal. When all N descriptions are available, the highest quality is attained. Several MDC techniques have been proposed, in particular for video [5]. A first class of solutions, called *channel splitting techniques*, are based on a suitable sampling of the original signal (*e.g.*, temporal sampling: odd images are encoded in the first description and even images in the second one) and an associated method for recovering missing samples [6, 7]; other techniques include non-conventional quantisers [8, 9], progressive coding with unequal protection [10], redundant and correlating transforms [11] (which enable scalability in the case of wavelets [12]), or insertion of redundant pictures to reduce error propagation [13]. However, we refer the reader to the excellent survey papers by Goyal [4] and by Wang *et al.* [5] for further information.

MDC can be used to improve robustness in video diffusion over a MANET by sending each description on a different and independent (logical or physical) channel. If at least one description is received, it can be decoded by a so-called *side decoder*, producing a relatively low-quality version of the original signal. If more descriptions (or all of them) are received, they are jointly decoded by a so called *central decoder*, achieving a higher quality. For the sake of completeness, it must be said that MDC is not the only technique used to adapt a video stream to a lossy channel: alternatives include cross-layer overlay construction with hierarchical video coding [14], joint use of scalable coding and advanced forward error correction codes such as Raptor codes [15], or, more recently, Network Coding (NC) [16, 17]. We notice that NC can be used in addition to multiple description coding (see Sec. V).

As for the problem of streaming multimedia content in MANETs, the literature provides a fair number of solutions.Studies on optimisation of content delivery on MANETs showed that structured protocols – *i.e.*, protocols that construct and maintain an overlay, such as Chord [18] – have a reduced delay and a better scalability, but a decreased resiliency, if compared to unstructured protocols such as Gnutella [19]. Structured protocols are more effective, but their performance degrades in more dynamic networks [20], which prevents interesting applications such as battlefield awareness services, out-of-office client briefing, or live delivery of special features at music or sport events. To overcome the limits of structured protocols in MANETs, we moved towards a cross-layer paradigm, which has been pointed out as a needed shift of perspective in protocol design for wireless transmission [21].

Authors are with Institut Mines-TELECOM, TELECOM-ParisTech, CNRS LTCI, 46 rue Barrault, F-75634 Paris, FRANCE. E-mail: {greco,cagnazzo,pesquet}@telecom-paristech.fr.

In this spirit, we recently proposed [22] a novel protocol, *A Broadcast Content Delivery Protocol* (ABCD), inherently designed for the MDC ad-hoc wireless case, exploiting the natural broadcast property of the medium in a cross-layered fashion. In most scenarios, ABCD performs well in terms of availability, robustness, scalability, and presents a low and stable latency. However, in large dense networks it is not able to abide by a stringent low-delay requirement. This is a common problem in the context of video streaming over MANETs, due to the fact that, even under optimal assignment of transmission ranges and traffic patterns, the throughput of each node in a wireless network diminishes to zero as the number of users is increased [23].

In this work, we propose a congestion control framework for real-time multiple description video multicast over MANETs. This framework includes models for congestion and distortion that take into account both the video stream coding structure and the unavoidable redundancy of the overlay network; it also provides the MAC layer with video-coding awareness, thus making possible to perform an optimisation of congestion and distortion. This framework can be integrated into any tree-based video streaming protocol for MANETs to improve its performance; we show here that, if integrated into the ABCD protocol, it attains a significant reduction of both average (over time and nodes) and maximum end-to-end delay, maintaining a delivery rate close to $100\,\%$.

As we write, several solutions for video streaming over ad-hoc networks have begun to appear, *e.g.*, based on centralised overlay construction supervised by the video source [24], or on simple loss-distortion models for one-to-one communication [25]. However, our technique still distinguishes itself from the others on some unique features. First, the application driven broadcast reservation, which allows the exploitation of the broadcast medium properties in an efficient way for one-to-many applications. Moreover, ABCD forms its independent multicast trees in a distributed fashion. Finally, our cross-layered congestion control technique, wherein optimisation is both application and network aware, includes a distortion model that takes into account both the coding structure and the estimated number of nodes affected by a loss. The idea of congestion-distortion optimisation (CoDiO) in video streaming, as opposed to the traditional rate-distortion optimisation, was introduced to model the effects that self-inflicted network congestion has on video quality [26]. The scenario was a wired network where each node is connected to the video source by a succession of high-bandwidth shared links and terminating with a bottleneck on the last hop; the case of unicast streaming over MANETs was considered some time later [27], but the model provided neither an on-line estimation of the network conditions, nor a viable extension for multicast streaming.

The ABCD protocol shares some key design principles with real-time peer-to-peer protocols for video streaming on mesh networks, such as VidTorrent [28] and its more scalable evolution SEACAST [29]. In fact, ABCD also aims to provide an adaptive self-repairing overlay forest, composed of independent trees each carrying a description of the stream. The wireless scenario, however, presents a different set of challenges, such as the increased need for decentralisation. Moreover, our work focuses on spontaneous multi-hop networks, wherein the video source is unable to collect in advance information about the number and the features of the nodes requiring the stream, and thus solves a class of problems different from those solved by rate-less codes. However, the two methods are not incompatible: if the MDC stream is also scalable, a technique of unequal loss protection via sliding-window Raptor codes [30] could be integrated in the per-hop transmission of nodes, providing resiliency to fading and shadowing.

The main contribution of this work is a CoDiO framework based on a distributed estimation of both the network topology, in order to capture the multiple paths that a video packet may follow, and the channel conditions, in order to estimate the effects on end-to-end delay. This information is propagated in an efficient and compact way through the network, leading to significant improvements in terms of both delay and objective video quality, as demonstrated by the simulations.

The reminder of this paper is organised as follows: in Sec. II, we resume the ABCD protocol, while in Sec. III, we describe the proposed CoDiO approach. Sec. IV presents our experimental setup and the simulation results. Finally, in Sec. V, general observations and an outlook on future work are given.

II. The ABCD protocol

The ABCD protocol [22, 31] was introduced to enable the construction of an overlay network composed of N application-level multicast trees, one for each description. Our system consists in a multitude of mobile nodes connected by wireless links in a mesh topology; the nodes are placed randomly in a fixed-size playground, and free to move within it. Since our application is aimed at military and rescue operations, we assume that the nodes move at walking/running speed, *i.e.*, around $5\,$m/s. Also, nodes can connect and disconnect abruptly. The application aims at delivering a video stream – available at a pre-selected node (*source*) – to the remaining nodes in the network (*peers*), and no other significant traffic is present on the MANET. The goal of the protocol is to produce an overlay that is efficient and robust. Efficient here means that the stream is delivered to all nodes with the minimum use of resources; robust, that the overlay is not severely affected by burst packet losses, due to collisions, mobility, or abrupt disconnection of nodes. It is understood that each node aims at receiving as many descriptions as possible, in order to maximise its video quality. Building and maintaining an overlay inevitably requires that a number of packets is exchanged; in order to reduce this number, nodes have to gather information without making explicit requests, by inferring as much as possible from any packet they received.

We exploit the fact that the wireless medium is inherently broadcast, hence each node can intercept any packet sent within its transmission area, as long as it does not collide. However, using the wireless channel as a broadcast medium conflicts with the fact that the 802.11 MAC layer was mainly designed for one-to-one communications, while its support of one-to-many communication is known to be unreliable and inefficient [32]. To overcome this difficulty, we provided ABCD with a modified MAC layer, better suited to our purposes; namely, we implemented a form of application-driven reliable broadcast [33]. The MAC is modified for ABCD video packets as follows. We set the destination field of the 802.11 frames to the broadcast address, which allows any ABCD video packet to be received by any *neighbour* of the sender. We define as neighbours of a node n all the nodes whose packets can be received by n; if we assume a symmetric channel, the node's neighbourhood is the set of nodes within its transmission area. Note that the neighbourhood of a node may change over time due to mobility and churn. Even though packets are sent in broadcast, we do enforce an RTS/CTS/ACK exchange with one neighbour, specified by the protocol accordingly with the application logic, and referred to as *control peer*.

The choice of the control peer, as discussed in our previous work [31], is implemented as a biased random choice. For each packet to be sent in broadcast, the control peer selection procedure is repeated, and each neighbour n of the sender s is selected with a probability $p_{\mathrm{CP}}(n) = \frac{T(n)+1}{T(s)}$, where $T(x)$ is the number of descendants of node x on the overlay tree (excluding x itself). In other words, $p_{\mathrm{CP}}(n)$ is the fraction of descendants of node s belonging to the sub-tree rooted in n (including n itself). Even though this technique cannot entirely prevent collisions, experimental evidence suggests that it reduces this phenomenon to the point of being negligible in our simulations, up to a density of 20 nodes per neighbourhood. Such a reduction of the collision probability allows the protocol to perform better than the standard 802.11 in the trade-off between rate and diffusion area. This trade-off is typical of self-limiting multi-hop broadcast in wireless networks [34], where a greater number of retransmissions increments the spread of the content (diffusion area), but reduces its throughput (rate) because of the limited channel capacity. Our improvement comes at the price of an increased congestion, as the channel reservation has an overhead in terms of time needed to transmit a packet; so it is advisable to use this technique only when a relatively small subset of nodes is transmitting at the same time, possibly with a high bitrate. Also, in order to work efficiently, the reliable broadcast needs a good choice of the control peer, which cannot be made using MAC layer information only. In conclusion, this technique is well fit for cross-layer protocols dealing with large chunks of data, such as ours, but would not be advisable as a general purpose addition to the standard MAC layer.

Once a reliable channel for broadcasting is available, the protocol design is quite intuitive. As soon as the video source has a content to deliver, it sends an advertisement message. The source's neighbours, receiving that message, reply with an attachment message for each description. Attachment messages are interpreted as subscriptions to the description, so as soon as the stream has at least one subscriber, the source *activates*, *i.e.*, it starts broadcasting the video packets for that description. We define a node *active* on a description when it is transmitting that description. Conversely, we say that a node *deactivates* on a description when it stops broadcasting it, willingly or otherwise. The subscribing nodes – that we define as the source's *children* – keep sending periodical attachment messages to their *parent* (the source) in order to keep it active: the source deactivates when nodes turn off or get out-of-range. Each node that is not in the source's neighbourhood, but that is in the neighbourhood of at least one of its children, becomes aware of the availability of the descriptions as it receives its *peers*' attachment messages (a peer is any node other than the source). It then chooses one of these peers as its parent and sends it an attachment message; the node thus chosen will activate, starting to forward the video packets it receives from the source for the description it is active on. The attachment messages sent by the newly subscribed node will now advertise the description within their neighbourhood, generating other subscriptions; this process is reiterated, independently on each description, until all nodes have one parent per description. A node can have a different parent for each description; the overlay is thus formed of the superposition of N different trees. In conclusion, a node becomes aware of a path to the video resource as it intercepts an attachment message. Quite often, it actually intercepts attachments from multiple peers (either piggybacked in video packets from an active node or standalone from a subscribed node), and has to decide which peer provides the best path. Even if the node already has a parent, it could become aware of a better path, created by the connection or the mobility of a peer. Therefore, nodes need a metric for the paths through their neighbours, in order to choose the best one. To this end, we designed the following metric, which takes into account the above discussed objectives, that each node evaluates over all candidate parents (*i.e.*, nodes of whom it intercepted an attachment message), then selecting the one that minimises its value:

$$J = \omega_{\mathrm{h}} h + \omega_{\mathrm{a}} a + \omega_{\mathrm{d}} d - \omega_{\mathrm{g}} g - \omega_{\mathrm{q}} q, \tag{1}$$

where h is the number of hops to the source, a is the number of active peers in the node's neighbourhood, d is the number of descriptions (other than the current one) for which the node is already subscribed to the candidate parent, g is the number of peers subscribed to the same candidate parent, q is the average signal-to-noise ratio of the link to its parent, and the ω values are a set of positive real weights, chosen experimentally so that the average PSNR of the video sequences decoded by the nodes is maximised. Experiments show that these weights need not to be adjusted at run-time, as the optimisation is quite robust with respect to their choice, in the sense that, using values slightly different from the optimal ones, the protocol still attains good results in terms of received frames and video quality.

Hop count minimisation should always be preferred over all other parameters in the function, since it assures that the overlay graph is acyclic (*i.e.*, a proper tree). Also, it is beneficial to the minimisation of the end-to-end delay. As a result, in an overlay generated by ABCD, a node cannot have, in a steady state, a peer in its neighbourhood whose hop-count is smaller than that of its current parent, since in that case it would simply switch parent in order to prevent loops. The number of active nodes per neighbourhood is also minimised, for two reasons: reducing the number of packets injected in the network (hence the congestion) and reducing the total amount of resources demanded to the nodes, which pay an energy cost to relay a description. The protocol also aims, by minimising $\omega_{\mathrm{d}}d$, at ensuring path diversity among the descriptions, which is advisable for both fairness and robustness. Notice that the term $-\omega_{\mathrm{g}}g$ implies that the nodes try to maximise the number of peers subscribed to their same parent (*siblings*), in order to concentrate subscriptions on fewer active nodes, making deactivations more frequent. As a result, the overlay trees generated by ABCD tend to be short and wide, and the number of active nodes tends to be small, which helps in preventing collisions.

However, we implemented a number of techniques explicitly aimed at reducing the collision probability. To make it unlikely that two (or more) active siblings rely a packet at the same time, video or attachment, we used a random assessment delay (RAD, [33]). To reduce the collision probability among video packets belonging to different descriptions, the source relays them as temporally far apart as possible. In particular, let us indicate with $t_{\mathrm{r}}(\ell, j)$ the time when the ℓ-th packet of the j-th description is relayed, and with T the data packet period. We consider the most common case of $N = 2$ descriptions, see also Sec. III-A. If the ℓ-th packet of description 0 is relayed at time $t_{\mathrm{r}}(\ell, 0) = t_0 + \ell T$, then we impose that the ℓ-th packet of description 1 is relayed at time $t_{\mathrm{r}}(\ell, 1) = t_0 + (\ell + 1/2)T$. Even though this "phase opposition" is lost when the two descriptions are propagated through the network, since they are not assured to arrive with the same delay to a generic node (because of the RAD and other random events), the difference in delays between the descriptions (*jitter*) is typically much smaller than the inter-packet semi-period $T/2$, hence collisions between packets of different descriptions are kept very unlikely. If more than two descriptions are used, we set $t_{\mathrm{r}}(\ell, j) = t_0 + (\ell + j/N)T$. Since N is usually not large [5], the delay difference is still much smaller than T/N. Finally, to mitigate collisions involving attachment messages, the attachment rate is reduced as a function of the number of siblings (including the hidden ones, as the parent node piggybacks the number of known children in its video packets: in steady state, active nodes tend to have a perfect knowledge of the number of their children). In conclusion, the impact of collisions is almost null in all our simulations.

As a final remark, we underline ABCD is a cross-layer protocol covering the MAC, network, and application layers, but not the physical layer: to this end, we build on the features of 802.11. Nevertheless, we are aware that physical layer issues must be taken into account when validating ABCD. For this reason, we have extensively tested it using a suitable simulation environment (*ns-2*) under several conditions of node density, number of nodes, and stream bitrate [22]. In all these scenarios, the protocol has proven to be able to ensure that $100\,\%$ of the nodes receive almost all frames of all descriptions, for a node density up to 20 nodes per neighbourhood, which is three times as high as the optimal density (in the sense of the trade-off between the number of hops to reach a destination and the collisions occurring at each node) [35]. The average delay is kept in the order of the hundreds of milliseconds as the topology is slowly changing, but the maximum delay can have much higher peaks if the topology is changing quickly, *e.g.*, a flash-crowd or a high mobility happens.

III. Congestion Distortion Optimisation for ABCD

Even though the ABCD protocol is able to create a dynamic overlay that is efficient in terms of coverage and packet overhead, it has an inherent limitation due to the broadcast reservation: when node density is very high or a sudden change in topology occurs, the average delay may become so high that some video frames are received beyond their playback deadline. The actual values for an acceptable delay depend on the buffering strategy and, more generally, on the application; in the following we shall assume that our target application aims at abiding by a conversational pattern, which means that the maximum accepted delay from the video source to the end user is in the order of one hundred milliseconds, and the total (*i.e.*, for all descriptions) bitrate of the stream is in the order of a few megabits per second, a setup consistent with a video-conferencing application. An example of application is a field awareness or an order dispatch service, provided in a military or disaster-relief scenario, where a unidirectional video feed can be provided live to the agents on the field by the central headquarters.

To reduce the delay in ABCD, we introduce a Congestion *vs.* Distortion Optimisation (CoDiO) criterion in the per-hop forwarding of the protocol; namely, we adjust the retry limit used by the RTS/CTS mechanism of the MAC, in a Co-Di optimised fashion. CoDiO is an approach already proven viable in the design of cross-layer protocols for video streaming on MANETs [36]; here we propose a formulation of the problem that takes into account both the inherent redundancy of the overlay and some specific features of the reliable broadcast scheme of ABCD; however, the proposed framework lends itself to be integrated in other tree-based streaming protocols.

We start from the observation that the congestion *vs.* distortion trade-off can be adjusted by tuning the retry limit k in the RTS/CTS mechanism. Small values of k would reduce the congestion, since less requests are sent to try and obtain the channel, but the expected distortion would increase, as it would increase the probability of not obtaining the channel, thus being unable to send the current packet. On the other hand, higher values of k would lower the expected distortion, since the probability of sending the packet is higher, but would also imply a higher congestion due to the channel occupation. We end up with a constrained minimisation problem; specifically, for each video packet, we want to find the optimal value k^* for the retry limit, defined as:

$$k^* \triangleq \arg\min_{\mathrm{k}\in\mathbb{N}} \left\{ J(k) \right\}, \text{ with } J(k) = D(k) + \lambda C(k), \qquad (2)$$

where $D(k)$ is the expected total distortion over the set of frames depending on the current packet (*i.e.*, contained in the packet or predicted upon it), for all the nodes in the sub-tree rooted in the current node (described in Sec. III-A), and $C(k)$ the expected congestion of the channel seen by the current node (detailed in Sec. III-D), both resulting from the retry limit k for the current packet. The parameter λ can be determined experimentally, as detailed in Sec. IV.

While congestion can be computed locally without the need to propagate information through the overlay, since it depends on the channel that the nodes can observe directly, the distortion model offers several challenges. A missing packet affects in general several frames of the decoded video sequence; moreover, the effects may be different for each node, depending on its reception of the same packet (from another path) or of the other description. Finally, losses propagate along the multicast tree, while a node can only communicate with its direct neighbours. These points are detailed in the following.

A. Distortion Model

Here, we present our distortion model and discuss how its parameters can be estimated and propagated. For the sake of simplicity, we shall assume that the stream consists of two descriptions (d_0 and d_1), hence the overlay consists in the superposition of two different multicast trees. The task of extending the framework for more than two descriptions is conceptually easy, but using more than two descriptions would require that *any* combination of received descriptions should be considered individually, making the exposition long and confused. As a matter of fact, the case of double description coding is the most interesting in practice, and in our reference scenario in particular, as it provides a very good balance between redundancy introduced in the stream by MDC, and robustness to losses [37, 38].

Let us consider a node n that has to send a packet r from description d_0. Each description can be decoded independently, but – within the same description – some frames could depend on previously decoded ones if a predictive scheme is employed. Let us call $\mathbf{I}(r)$ the set of video frames that depend on r because they are included in r or predicted upon it. We assume an additive distortion measure, since all the most popular distortion measures are additive or equivalent to additive measures: *e.g.*, the sum of squared differences, SSD, equivalent to the PSNR; the sum of absolute differences, SAD; or the structural similarity (SSIM) index [39]. Using any of these measures (in particular, we used the SSD), we can define the following quantities: D_c, the cumulative distortion for $\mathbf{I}(r)$, *i.e.*, the sum of the distortion of the frames in $\mathbf{I}(r)$, when the central decoder can be used on these frames, since both descriptions have been received; D_0, the cumulative distortion for $\mathbf{I}(r)$ when decoding only d_0 (only r is received); D_1, the cumulative distortion for $\mathbf{I}(r)$ when decoding only d_1 (only r is lost); and D_f, the cumulative distortion for $\mathbf{I}(r)$ when using a strategy of concealment (as both d_0 and d_1 have been lost).

Let us consider what happens if node n is *deactivated* on description d_0, *i.e.*, it stops transmitting it. Each node in the sub-tree rooted in n belongs to exactly one of the following sets.

- $\mathbf{S}_\text{c}$, of size N_c: nodes able to receive both descriptions even if node n is deactivated on d_0;
- $\mathbf{S}_0$, of size N_0: nodes able to receive only d_0 if node n is deactivated on d_0;
- $\mathbf{S}_1$, of size N_1: nodes able to receive only d_1 if node n is deactivated on d_0;

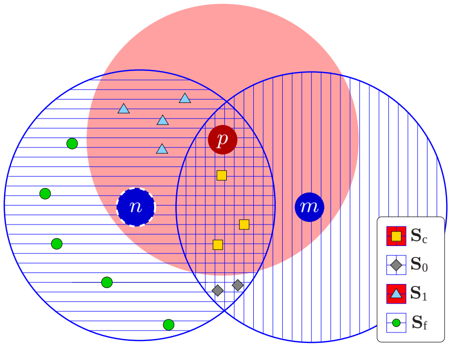


Figure 1. Sets $\mathbf{S}_\text{c}$, $\mathbf{S}_0$, $\mathbf{S}_1$ and $\mathbf{S}_\text{f}$ with respect to node n. Here $N_\text{c} = 3$, $N_0 = 2$, $N_1 = 4$ and $N_\text{f} = 5$. Only nodes n, m and p are active. The pattern in the background of the nodes' transmission area is parallel lines for description d_0 and solid for d_1.

- $\mathbf{S}_\text{f}$, of size N_f: nodes unable to receive either description if node n is deactivated on d_0.

In Fig 1, we show a node n (identified by the blue dot with dashed border) transmitting description d_0 in its neighbourhood, which in this example is a circle. It should be noted that the actual shape of the transmission area does not affect the protocol nor the definition of the sets $\mathbf{S}_\text{c}$, $\mathbf{S}_0$, $\mathbf{S}_1$, and $\mathbf{S}_\text{f}$. Node m is also active on d_0 and is not a child of n; while node p is active on the complementary description d_1; the transmission areas of these nodes are also represented. If we assume that there are no other active nodes in this neighbourhood, considering the intersections of the transmission areas, we see that n has three neighbours in its set $\mathbf{S}_\text{c}$ (identified by squares), two nodes belong to set $\mathbf{S}_0$ (diamonds), four nodes belong to set $\mathbf{S}_1$ (triangles); and five nodes belong to set $\mathbf{S}_\text{f}$ (circles). Even though this is easily seen on this simplified example, determining the size of these sets is much more complicated when we take into account a sub-tree rooted in n, as discussed below. Knowing the values of N_c, N_0, N_1, and N_f, we would be able to compute $D(k)$ perfectly. Unfortunately, in the reference ABCD protocol, n is not aware of which nodes in its sub-tree belong to which set; we show here how it is possible to estimate how many nodes belong to each set through the up-tree propagation of attachment messages.

For each video packet r, let us define $\eta_n(k)$ the delivery ratio for n given k as a retry limit, *i.e.*, the expected number of nodes in n's sub-tree receiving r if sent by n with k as a retry limit, normalised by the total number of nodes in the sub-tree. In the following, we shall omit the subscript identifying the node when unambiguous in the context. The delivery ratio models the fact that not all nodes in the sub-tree rooted in n will receive r, since we are dealing with a wireless (therefore lossy) network. We can define the distortion for a set of nodes as the sum of the distortions of the nodes in the set. The distortion for sets $\mathbf{S}_\text{c}$ and $\mathbf{S}_0$ does not depend on the reception of r sent by n, since the nodes in those sets would still receive a copy of r and decode description d_0, and shall therefore be omitted, as it plays no role in the optimisation. For set $\mathbf{S}_1$, only the nodes receiving r could decode both descriptions; the others would have to decode only description d_1. Similarly, the nodes in $\mathbf{S}_\text{f}$ receiving r could decode at least d_0, while the others would have to use a concealment strategy. Therefore, if we make the assumption that the delivery ratio for r is the same on both $\mathbf{S}_1$ and $\mathbf{S}_\mathrm{f}$ (which is reasonable, since the two descriptions are sent independently), the two contributions to the total distortion are $D_{\mathbf{S}_1} = N_1[\eta(k)D_\mathrm{c} + (1-\eta(k))D_1]$ and $D_{\mathbf{S}_\mathrm{f}} = N_\mathrm{f}[\eta(k)D_0 + (1-\eta(k))D_\mathrm{f}]$. We rewrite the sum of these two contributions as:

$$\begin{aligned} D_{\mathbf{S}_1} + D_{\mathbf{S}_\mathrm{f}} =& N_1 D_1 + \eta(k) N_1 \left(D_\mathrm{c} - D_1\right) + \\ &+ N_\mathrm{f} D_0 + (1-\eta(k)) N_\mathrm{f} \left(D_\mathrm{f} - D_0\right). \end{aligned}$$

We now introduce $\Delta D_\mathrm{c} = D_1 - D_\mathrm{c} > 0$ and $\Delta D_\mathrm{f} = D_\mathrm{f} - D_0 > 0$, which measure the reduction in the distortion a node experiences when switching from decoding d_1 only to decoding both, and from decoding no description (concealment) to decoding only d_0, respectively. These quantities are positive, since central decoding always outperforms side decoding and side decoding always outperforms concealment in terms of distortion [4, 5]. Using these quantities, we rewrite the total distortion as:

$$\begin{aligned} D(k) = [1-\eta(k)] N_\mathrm{f} \Delta D_\mathrm{f} - \eta(k) N_1 \Delta D_\mathrm{c} \\ + N_\mathrm{c} D_\mathrm{c} + N_1 D_1 + N_\mathrm{f} D_0 + N_\mathrm{f} D_\mathrm{f}, \end{aligned}$$

where $N_\mathrm{c}D_\mathrm{c}$, $N_1 D_1$, $N_\mathrm{f} D_0$ and $N_\mathrm{f} D_\mathrm{f}$ do not depend on k, and can be neglected in the optimisation.

In order to solve the minimisation problem, a node has to estimate the remaining contribution to the total distortion: the differences ΔD_c and ΔD_f, the delivery ratio $\eta(k)$, and the group sizes N_1 and N_f. Here, ΔD_c and ΔD_f depend on the codec used; they can be easily computed at the encoder – where both central and side distortion are known, and concealment distortion can be measured – and included in the video stream as headers, or they can be estimated, such as in our experimental setup. There exist simple and effective distortion models allowing a recursive estimation of the frames distortion [40–42].

Estimating the delivery ratio and the group sizes is a more challenging task, as this information is distributed and time-varying. In order to have a reliable estimation, nodes have to deduce, at least partially, the topology of the overlay beyond their transmission area (wherein they are able to collect information directly). In the following, we present a solution to this problem that relies on a small number of messages exchanged, sent through the same links used for the video.

B. Group size estimation

In order to solve problem (2), nodes have to estimate the group sizes N_f and N_1. To this end, we introduce a model of our ad-hoc wireless network as a simple directed graph $\mathbf{G} \triangleq (\mathbf{E}, \mathbf{V})$, where the vertices $n_i \in \mathbf{V}$ represent the mobile nodes and the edges $e_{ij} \in \mathbf{E}$ the wireless links between two nodes n_i and n_j, *i.e.*, if $e_{ij} \in \mathbf{E}$, n_i and n_j are neighbours (with respect to definition given in section II). Note that edges in $\mathbf{E}$ are bidirectional: $e_{ij} \in \mathbf{E} \Leftrightarrow e_{ji} \in \mathbf{E}$.

An overlay produced by ABCD will converge, for each description, to a directed tree $\mathbf{T}$, rooted in the source (labelled s), and spanning from $\mathbf{G}$. For the sake of simplicity, we shall hereby consider only a snapshot of the overlay topology at the time the packet has to be sent: $\mathbf{T} = (\mathbf{P}, \mathbf{V})$, where $\mathbf{P} \subset \mathbf{E}$ represents the links over which the video stream is sent, *i.e.*, the *parent-child* relation defined in Sec. II.

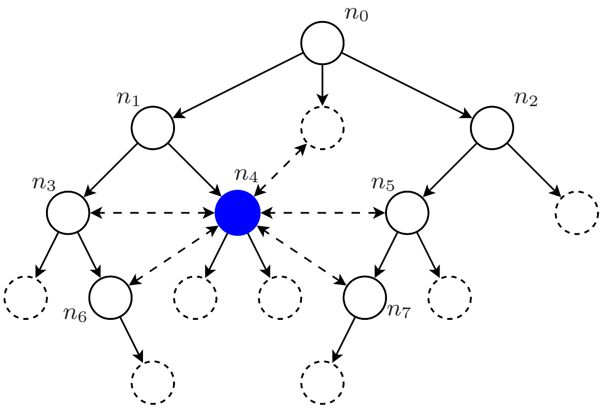


Figure 2. Example of directed graph of an ABCD overlay (all nodes, solid lines) for one description. Solid border represents active nodes. Neighbouring relations for node n_4 (dashed lines) are also represented. Here, $\mathcal{B}_4 = \{n_5, n_6, n_7\}$, $\mu(n_4, n_5) = \mu(n_4, n_7) = n_0$, and $\mu(n_4, n_6) = n_1$. Being $h(n_0) = 0$ and $h(n_5) < h(n_7)$, we designate as foster parent for n_4 the node n_5.

n_i	n_7	n_6	n_5	n_4	n_3	n_2	n_1
δ_i	n_0	n_1	n_0	n_0	n_1	n_0	n_0
c_i	2	2	4	3	4	6	8
x_i	0	0	0	0	0	0	0
φ_i	n_4	n_4	n_4	n_5	–	–	–

Table I
DEPENDENCY RECORDS GENERATED BY THE ACTIVE NODES IN FIG. 2 AND DESIGNATED FOSTER PARENTS.

We define now a set of useful relations between two nodes n_i and n_j on the ABCD overlay by using binary relations on $\mathbf{G}$: n_i and n_j are *neighbours* (which we denote by $n_i \curlyvee n_j$), if there exists in $\mathbf{E}$ a pair of edges e_{ij} and e_{ji} connecting them; n_i is the *parent* of n_j ($n_i \prec n_j$), if there exists in $\mathbf{P}$ an edge e_{ij}; conversely n_i is a *child* of n_j ($n_i \succ n_j$), if there exists in $\mathbf{P}$ an edge e_{ji}; n_i and n_j are *siblings* ($n_i \sim n_j$), if there exists in $\mathbf{V}$ a node n_k such that there exist in $\mathbf{P}$ two edges e_{ki} and e_{kj}, *i.e.*, they have the same parent. We also define n_i as an *ancestor* of n_j ($n_i \ll n_j$) if there exist a simple directed path from n_i to n_j, in which case we also define n_j as a *descendant* of n_i ($n_j \gg n_i$). Let us also denote by $h(n_i)$ the number of hops that a packet sent by the source has to cross in order to be received by n_i, *i.e.*, the value h used by node n_i in the objective function (1).

In order to estimate N_f and N_1, we analyse the impact of a node skipping the transmission of a packet. We introduce in the protocol the notion of *foster parent*: the idea is to have each node collect information about possible alternative paths from which the description can be received, then transmit it to its current parent, thus allowing the parent to estimate the impact of its decision of not sending a packet. This is a complex problem, as no node has a global view of the topology (nor it should, as propagating the whole topology to all active nodes each time it changes would congest the channel). Our contribution consists in a technique to reliably estimate the group sizes using the nodes' local information plus a small amount of information propagated through the overlay. Let us call $\mathbf{V}_{\mathrm{A}} \subseteq \mathbf{V}$, the set of active nodes, *i.e.*, the nodes that are relying at least one description, and are therefore parents of at least one node; for each node n_i, we define its set of *candidate foster parents* $\mathcal{B}_i \subset \mathbf{V}_{\mathrm{A}}$ as:

$$\mathcal{B}_i \triangleq \{\; n_j \in \mathbf{V}_{\mathrm{A}} \mid \\ (n_i \curlyvee n_j) \wedge (n_i \not\ll n_j) \wedge (n_i \not\gg n_j) \wedge (n_i \nsim n_j)\}\,,$$

which is the set of active neighbours of n_i that are neither ancestors, nor descendants, nor siblings of n_i (*e.g.*, in Fig. 2, $\mathcal{B}_4 = \{n_5, n_6, n_7\}$). Each node has perfect knowledge of its set of candidate foster parents. In fact, since candidate parents are both neighbours and active, the node receives their attachment messages piggybacked in their video packets, and from these it can infer their parent. Comparing its neighbours' parent with its own, it can deduce whether they are siblings. Furthermore, each node obviously knows both its parent and children. Finally, as mentioned is section II, it cannot have in its neighbourhood other ancestors than its parent (and, by symmetry, no descendant other than its children).

The existence of a node $n_j \in \mathcal{B}_i$ assures that n_i would stay connected even if the link with its parent were removed. Also, it is possible that it would stay connected even if the link between its parent and its grandparent were removed: this would hold true as long as the grandparent of n_i were not an ancestor of n_j. In order to evaluate the degree of robustness of the path that n_j provides to n_i, we want to investigate the *common ancestors* the two share. To this end, let us observe that any couple of nodes in $\mathbf{V}$ has at least one common ancestor, *i.e.*, the source. However, nodes can have more common ancestors, *e.g.*, two siblings share a number of ancestors equal to their height in the tree. Let us define:

$$\mathcal{C}_{ij} \triangleq \{n_k \in \mathbf{V}_{\mathrm{A}} \mid (n_k \ll n_i) \wedge (n_k \ll n_j)\}\,,$$

which is the set of common ancestors of n_i and n_j (*e.g.*, in Fig. 2, $\mathcal{C}_{4,6} = \{n_1, n_0\}$). In this set, we can identify the *most recent common ancestor* μ_{ij} as $\mu(n_i, n_j) \triangleq \arg\max_{\mathrm{n_k \in \mathcal{C}_{ij}}} \{h(n_k)\}$, which is the common ancestor of n_i and n_j with the longest path from the source (*e.g.*, in Fig. 2, $\mu_{4,6} = n_1$). Let $n_{\mathrm{c}} = \mu_{ij}$; any of the $h(n_{\mathrm{c}})$ ancestors of n_{c} will also be in C_{ij}, and any of its descendants will not. In other words, we can assume that, from n_{c} to n_i, the two paths (one through its current parent, the other through n_j), are affected by independent failures, *i.e.*, no failure of a single node between n_{c} and n_i could severe both paths. Therefore, we are interested in the neighbour n_j such that μ_{ij} is as close as possible to the source. This neighbour is called *designated foster parent* and denoted by φ_i. More formally:

$$\begin{cases} m_i & \triangleq \quad \min\limits_{n_j \in \mathcal{B}_i} h\left(\mu(n_i, n_j)\right), \\ \mathcal{M}_i & \triangleq \quad \{n_j \in \mathcal{B}_i \mid \mu(n_i, n_j) = m_i\}\,, \\ \varphi_i & \triangleq \quad \arg\min\limits_{\mathrm{n_j \in \mathcal{M}_i}} \{h(n_j)\}\,. \end{cases}$$

The rationale behind this choice is to minimise the number of *critical* nodes, *i.e.*, nodes that would cause the failure of both regular and alternative paths if deactivated. In order to find φ_i, a node must be able to compute the number of ancestors it has in common with its neighbours; this can be done by comparison if each node adds in its attachment messages the sequences of its ancestors, which can be easily generated as it suffices that any active node adds its identifier to the sequence it receives from its parent. The amount of data exchanged to propagate the sequences is small, as the ABCD trees tend to be short and wide (see section II). It should be noted that, even in the original version of ABCD, upon disconnection of their current parent, nodes could still receive data from paths unsevered by the disconnection. However, this mechanism was implicit and, more importantly, the existence of alternative paths was not propagated through the overlay, therefore other nodes could not rely on this information to make any decision. Even though all alternative paths existing in the original protocol still exist, we *designate* one and advertise information about this designation to allow other nodes to benefit from it.

In the estimation of groups size, we shall assume that each node n_i designates a unique foster parent φ_i. The reason is that, even though in principle *any* node in $\mathcal{B}_i$ provides an alternative path, it is unlikely that in case of failure of both the regular and alternative path, due to the disconnection of a node in their common path, other paths in $\mathcal{B}_i$ could still be active, since the protocol tries to concentrate subscriptions on as few nodes as possible (see Section II), and if a node fails that is a common ancestor of n_i and φ_i, it is likely to be an ancestor of the others nodes in $\mathcal{B}_i$ as well.

The presence of an alternative path allows a node to receive a packet even though its parent decided not to send it or was unable to obtain the channel. These alternative paths determine the groups $\mathbf{S}_1$ and $\mathbf{S}_{\mathrm{f}}$, whose sizes (N_1 and N_{f}) we want to estimate. In order to do so, we need to spread the information about the existence of these alternative paths through the overlay tree. This information, however, has to be refined while it is spread from the leaves towards the source, in order to prevent congestion. The propagation of the information about alternative paths works as follows. First, each node n_i finds its most recent ancestor in common with its designated foster parent, called *path dependency node* and denoted δ_i. By convention, a node with no alternative path defines its current parent as path dependency node.

$$\delta_i \triangleq \begin{cases} \mu(n_i, \varphi_i) & \text{if } \mathcal{B}_i \neq \emptyset, \\ n_k \mid n_k \prec n_i & \text{otherwise.} \end{cases}$$

The meaning of δ_i is that n_i has an alternative path that is independent from the current path up to δ_i. See Fig. 2 and Tab. I for an example.

Once a node has computed its path dependency node, it has to transmit to its parent the information about the path dependencies. This is done using *dependency records* (see Tab. I), which we define as $\mathbf{d_i} \triangleq [\delta_i, c_i, x_i]$, where c_i is the number of nodes in the sub-tree rooted in n_i sharing the same dependency, and x_i is a flag signalling whether the nodes are also receiving the complementary description. In order to explain how dependency records work, we shall now describe how they are generated and propagated through the tree. Dependency records are generated by each leaf n_i as $\mathbf{d_i} = [\delta_i, 1, x_i]$, one for each description it receives, then sent it to its parent. The parent of n_i interprets this record as follows: for description d_0, n_i has an independent path up to δ_i and it is the only one in its sub-tree having this dependency node; also, it has (if $x_i = 1$), or has not (if $x_i = 0$), the complementary description d_1. Another similar dependency record is generated for d_1. Let us assume that n_j is the parent of n_i; if n_i has no alternative path or its path dependency node corresponds to the parent of n_j, then n_j updates the record, replacing the path dependency node of n_i with its own. The logic is simple: n_i stays connected as long as n_j is connected; if n_j has a path alternative to its current one, then it is able to restore the path from the source to n_i even if its current path fails. Therefore, for each child's dependency record $\mathbf{d_i}$, the parent node n_j generates an *updated dependency record*:

$$\mathbf{d_i}' \triangleq \begin{cases} \mathbf{d_i} & \text{if } \delta_i \neq n_j \text{ and } \delta_i \nprec n_j, \\ [\delta_j, c_i, x_i] & \text{otherwise.} \end{cases}$$

In other words, $\mathbf{d_i}'$ represents the dependency record of n_i, with the addition of the knowledge of the topology contributed by n_j: if n_j is the path dependency node for n_i, it can update this information.

Node n_j needs now to transmit an aggregated information about its own alternative paths and its children's. In order to do so, it generates two *aggregate dependency records*, one for the children receiving the complementary description, and the other for those not receiving it:

$$\mathbf{s_j}^{\mathbf{k}} \triangleq \sum_{i \mid n_i \succ n_j \wedge x_i = k} \mathbf{d_i}', \qquad \forall k \in \{0, 1\},$$

where the sum denotes the composition of two records having the same value of the flag x_i, defined as:

$$\begin{aligned} &[c_i, \delta_i, x_i] + [c_j, \delta_j, x_i] \triangleq \\ &[(c_i + c_j), \arg \min_{\delta_\mathrm{k} \in \{\delta_\mathrm{i}, \delta_\mathrm{j}\}} \{h(\delta_k)\}, x_i]. \end{aligned}$$

This rule of composition respects the semantic of the record: in fact, the number of depending nodes is obtained as the sum of the contribution of each record, while the path dependency node is chosen consistently with the definition of most recent common ancestor given above. In other words, in these records, the new path dependency node for the set of children sharing the same value of x_i is the one with the shortest height. Aggregation of dependency records is needed, as it transforms a series of local views into a more descriptive global information: n_j has to propagate its knowledge up-tree to facilitate its ancestor in making decisions that affect the whole sub-tree. Thus, at higher levels, a global optimisation is performed, using an aggregate information on the descendants. At lower levels, active nodes may or may not operate the same choice as their ancestors, since they will now use a more and more detailed local information. Finally, n_j will add its own contribution to the proper aggregate dependency record, in accordance with the value of x_j:

$$\begin{cases} \mathbf{d_j}^{\mathbf{x_j}} & \triangleq & \mathbf{s_j}^{\mathbf{x_j}} + [1, \delta_j, x_j] \\ \mathbf{d_j}^{\mathbf{1-x_j}} & \triangleq & \mathbf{s_j}^{\mathbf{1-x_j}} \end{cases}$$

Node n_j will then propagate both $\mathbf{d^0}_j$ and $\mathbf{d^1}_j$. Using the records received from its children, an active node n_i can estimate the group sizes as follows:

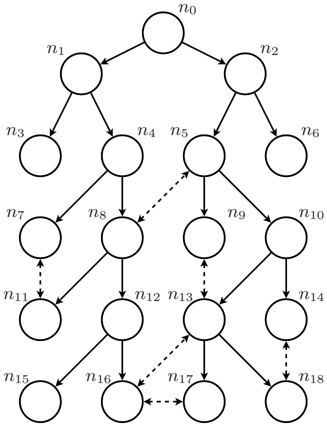


Figure 3. Example of directed graph of an ABCD overlay for one description (only active nodes are depicted). Neighbouring relations (dashed lines) are also represented, for couples of nodes not connected by a parent-child or sibling relation.

$$\begin{aligned} \widetilde{N}_\mathrm{c} &= \sum_{j \in \mathcal{C}_\mathrm{c}} c_j \quad \text{with } \mathcal{C}_\mathrm{c} = \{j \mid n_i \prec n_j \wedge \delta_j \ll n_i \wedge x_j = 1\}\,; \\ \widetilde{N}_1 &= \sum_{j \in \mathcal{C}_1} c_j \quad \text{with } \mathcal{C}_1 = \{j \mid n_i \prec n_j \wedge \delta_j \not\ll n_i \wedge x_j = 1\}\,; \\ \widetilde{N}_0 &= \sum_{j \in \mathcal{C}_0} c_j \quad \text{with } \mathcal{C}_0 = \{j \mid n_i \prec n_j \wedge \delta_j \ll n_i \wedge x_j = 0\}\,; \\ \widetilde{N}_\mathrm{f} &= \sum_{j \in \mathcal{C}_\mathrm{f}} c_j \quad \text{with } \mathcal{C}_\mathrm{f} = \{j \mid n_i \prec n_j \wedge \delta_j \not\ll n_i \wedge x_j = 0\}\,. \end{aligned}$$

In other words, n_i verifies, for each child n_j, if the dependency for the alternative path of n_j is satisfied, *i.e.*, if the path dependency node declared in the record is one of its ancestors (rather than n_i itself); if this is the case, then it is understood that c_j nodes in the sub-tree rooted in n_j are able to receive d_0 even if n_i is deactivated on that description. These c_j nodes are therefore accumulated – depending on the value x_j declared in the record – either on $\widetilde{N}_\mathrm{c}$ (if $x_j = 1$) or on $\widetilde{N}_0$ (if $x_j = 0$). On the other hand, if the path dependency node is not an ancestor of n_i (*i.e.*, it is n_i itself), then the alternative path is invalidated, and the c_j nodes are assumed to be unable to receive d_0 if n_i deactivates; they are therefore accumulated on either $\widetilde{N}_1$ (if $x_1 = 1$) or $\widetilde{N}_\mathrm{f}$ (if $x_1 = 0$).

In Tab. II we present an example of group sizes estimation, with respect to the overlay depicted in Fig. 3. For the sake of simplicity, let us assume that $x_i = 0$ for all nodes. The estimation of the group size differs slightly from the actual values (namely for n_4 and n_5, with an error of one unit), as nodes are unable to determine their common ancestor over *all* the paths of their designated foster parent. In the example, n_7 assumes n_4 as dependency node for its alternative path through n_{11}, therefore n_4 invalidates it when estimating the group sizes; however, even if n_4 deactivates, n_7 could still actually receive the description through the path $n_0 \rightarrow n_2 \rightarrow n_5 \rightarrow n_8 \rightarrow n_{11} \rightarrow n_7$, *i.e.*, through the alternative path of n_8. However, even with this simplification – which reduces the amount of information exchanged – the

n_i	n_{18}	n_{17}	n_{16}	n_{15}	n_{14}	n_{13}	n_{12}	n_{11}	n_{10}	n_9	n_8	n_7	n_6	n_5	n_4	n_3	n_2	n_1	n_0
φ_i	n_{14}	n_{16}	n_{13}	–	n_{18}	n_9	–	n_7	–	n_{13}	n_5	n_{11}	–	n_8	–	–	–	–	–
δ_i	n_{10}	n_0	n_0	n_{12}	n_{10}	n_0	n_0	n_4	n_0	n_5	n_0	n_4	n_2	n_0	n_0	n_1	n_0	n_0	–
c_i	1	1	1	1	1	3	3	1	5	1	5	1	1	7	7	1	9	9	19
$\widetilde{N}_0$	–	–	–	–	–	1	1	–	3	–	4	–	–	**5**	**5**	–	7	7	0
N_0	–	–	–	–	–	1	1	–	3	–	4	–	–	6	6	–	7	7	0
$\widetilde{N}_\mathrm{f}$	–	–	–	–	–	1	1	–	1	–	0	–	–	**1**	**1**	–	7	7	0
N_f	–	–	–	–	–	1	1	–	1	–	0	–	–	0	0	–	1	1	18

Table II

VALUES USED FOR ESTIMATION OF GROUPS SIZE IN FIG. 3. WE REPORT HERE THE DESIGNATED FOSTER PARENTS φ_i, THE FIELDS OF THE DEPENDENCY RECORDS, δ_i AND c_i ($x_i = 0$ FOR ALL NODES), THE ESTIMATED VALUES $\widetilde{N}_0$ AND $\widetilde{N}_\mathrm{F}$, AND THE ACTUAL VALUES N_0 AND N_f.

reliability of the estimation is not affected much, as we see in the example, where the small errors on the estimation are corrected upper in the tree. This is a point of strength of the protocol: errors do not propagate through the whole tree; there may be local errors, but they tend to be corrected as dependency records propagate, as they are enriched with new topological information.

C. Delivery ratio estimation

We shall now discuss how $\eta_n(k)$ is estimated. To this end we make the following assumptions:

1) Before each video packet is sent, the sender transmits an RTS message to the control peer.
2) If the sender receives a CTS message from the control peer, then it gains exclusive access to the channel and the video packet shall be received correctly by the control peer.
3) If the control peer has correctly received a video packet, to which it replies with an ACK message, then all designated receivers have correctly received the same packet; *i.e.*, when the sender receives an ACK from the control peer, it can be inferred that all the children received the data packet.

The first assumption is enforced by the ABCD protocol itself, the other two are common assumptions justified by the way the RTS/CTS/ACK mechanism of IEEE 802.11 works. In practice, there exists a minor fraction of nodes not receiving a video packet even if it has been acknowledged by its control peer, in certain topologies. However, these events are always limited in number of both nodes and packets, since the ABCD parent switch mechanism tends to avoid these topologies in the first place. Also, in a scenario with node mobility, these pathological topologies are necessarily transient. Of course, in a wireless environment, there is always the possibility that one or more descendants of a node do not actually receive the packet because of fading. However, on one hand 802.11 provides several tools to reduce this problem. On the other, the only effect that our assumption could have on the optimisation process is a slight overestimation of $D(k)$, which does not necessarily translate into a wrong selection of the retry limit, since the group size estimation can be affected by a small error of the opposite sign, and k is selected into a discrete and relatively small set, therefore small variations of D are drowned by the quantisation on k. Finally, the soundness of these assumptions is supported by experimental evidence both in the articles proposing broadcast reservation [33, 43] and in tests performed on ABCD itself [22].

Let us consider a node n that has at least one child but no grandchildren. We call p_n the probability of n obtaining the channel with a single try, which, under the assumptions made above, the node can estimate by just using its video packets as probes for the channel around itself, with an exponentially-weighted moving average of the number of received ACKs divided by the number of RTS sent. Node n estimates $\eta_n(k)$ as $1 - (1 - p_n)^k$, then sends the value of $\eta_n(1) = p_n$ to its parent. Note that here we are also assuming that if one child of node n receives a packet, all of the children of n also do, consistently with the assumption made above. Let us now consider a second node m that has at least one grandchild, *e.g.*, the parent of n; m will receive from each of its children c its delivery ratios η_c and the number of its descendants g_c. If with a single try m were able to obtain the channel, the message it would send could be received by its g_m children and, by inductive hypothesis, any node c child of m would reach in its turn $\eta_c(1)g_c$ of its g_c descendants. The value $\eta_m(1)$, can therefore be estimated as:

$$\eta_m(1) = p_m \times \frac{g_m + \sum\limits_{c \succ m} \eta_c(1) g_c}{g_m + \sum\limits_{c \succ m} g_c}.$$

This formula can be read as follows: if m obtains the channel (which happens with probability p_m), then the packet is received by its g_m children, plus (on average) $\eta_c(1)g_c$ descendants through each child c, out of the total number of its descendants ($g_m + \sum\limits_{c \succ m} g_c$). All other values of $\eta_m(k)$ are then estimated as $\eta_m(k) = 1 - [1 - \eta_m(1)]^k$.

The values of g_c are already part of the node state piggy-backed in attachment messages (see Sec. II) and the values of η_c can also be transmitted in the same way, so no congestion is generated in order to transfer this information. Also, piggy-backing in attachment messages assures us that the information is always up-to-date in case topology changes.

D. Congestion

With respect to the minimisation problem (2), we define the congestion seen by a node n as the number of packets Q_n that cannot be sent (by n or by other nodes) as the channel in the neighbourhood of n is occupied by n itself, times the time $\Theta(k)$ the channel is kept busy by n (the former not depending on k): $C(k) = Q_n\Theta(k)$. Here, Q_n is the length of a virtual packet queue, distributed among n and its neighbours. This is different from the assumption usually made in the literature that *all* the neighbours of a node are always willing to transmit at any time, and is based on the knowledge of the neighbours' state, namely, the number of packets they have to send — in line with the ABCD protocol paradigm of piggybacking control information on broadcast packets. Notice that, with this formulation, a node will not refrain from sending a packet for the mere fact of having many neighbours: its neighbours shall not be considered when it is known that they do not have a packet to send; conversely, the more a node has neighbours with pending transmissions, the more it will try to reduce the number of packets it sends.

We estimate the time the channel is kept busy by the quantity $\Theta(k) \triangleq T_{\text{rts}}(k) + P_n(k)T_{\text{pkt}}$, where T_{pkt} is the time to transmit the data packet — which depends on its size. The term $P_n(k)$ is the probability that n successfully obtains the channel – and thus sends the packet – with at most k tries, which is estimated as $P_n(k) = 1 - (1 - p_n)^k$. The value of $T_{\text{rts}}(k)$ is defined as the expectation of the random variable $t_{\text{rts}} = T_{\text{rts}}(k) \triangleq \mathbf{E}\left[\, t_{\text{rts}} \,\right]$, which represents the time needed to succeed the RTS/CTS competition, and which depends on the number c of collisions.

Let χ be the random variable representing the number of collisions occurred before obtaining the channel, t_{rts} can be written as $t_{\text{rts}} = (\chi + 1)\,T_{\text{tx}} + \chi\,t_{\text{bo}}$, where T_{tx} is the time to send an RTS packet including its following inter-frame space (as defined by the IEEE 802.11 standard [44]), and t_{bo} is the back-off time, also depending on χ. The dependency of t_{rts} on χ suggests to evaluate $T_{\text{rts}}(k)$ as a conditional expectation versus the number of occurred collisions:

$$\begin{aligned} T_{\text{rts}}(k) &= \sum_{c=0}^{k-1} \mathbf{E}\left[\, t_{\text{rts}} | \chi = c \,\right] \Pr\{\chi = c\} \\ &= \sum_{c=0}^{k-1} \left((c+1)T_{\text{tx}} + c\,\mathbf{E}\left[\, t_{\text{bo}} | \chi = c \,\right]\right) \Pr\{\chi = c\}. \end{aligned}$$

Given c collisions, the expected value of t_{bo} is $\mathbf{E}\left[\, t_{\text{bo}} | \chi = c \,\right] = \frac{W}{2}\,(2^c - 1)$, where W is the size of the contention window as defined by the 802.11 standard [44]. Therefore, given the probability of having c collisions $\Pr\{\chi = c\} = p_n(1 - p_n)^c$, $T_{\text{rts}}(k)$ can be written as:

$$T_{\text{rts}}(k) = \sum_{c=0}^{k-1} \left[(c+1)\,T_{\text{tx}} + c\,\frac{W}{2}\,(2^c - 1) \right] p_n(1 - p_n)^c.$$

We observe that, if we define:

$$\Delta T_{\text{rts}}(k) \triangleq \left[kT_{\text{tx}} + (k-1)\frac{W}{2}(2^{k-1} - 1) \right] p_n(1 - p_n)^{k-1},$$

then $T_{\text{rts}}(k)$ can be computed using the following difference equation:

$$T_{\text{rts}}(0) = 0, \qquad T_{\text{rts}}(k) = T_{\text{rts}}(k-1) + \Delta T_{\text{rts}}(k).$$

This is a very convenient formulation, as several consecutive values of $T_{\text{rts}}(k)$ have to be computed.

The congestion estimation completes the set of values needed to solve the minimisation problem (2). With all the parameters available, each node can decide its value of k for the current packet by simply evaluating $J(k)$ for all k up to a maximum value. The choice of this value, along with an extensive experimental validation of the CoDiO model and approach, can be found in the following section.

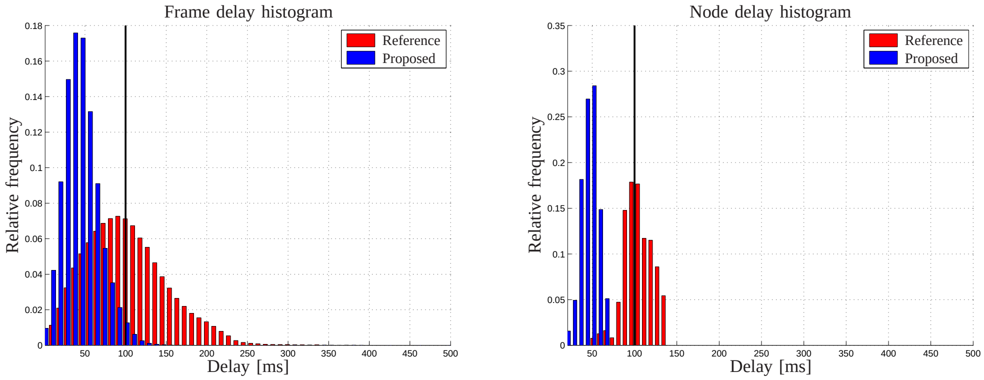
IV. Experimental results

In this section, we present the results of the performance tests of the proposed CoDiO extension of the ABCD protocol, in comparison with the conventional ABCD implementation.

A set of nine video sequences ("Akiyo", "Bus", "City", "Coastguard", "Flower", "Football", "Foreman", "Mobile", and "Stefan"; CIF at 30 fps), concatenated and looped to match the total simulation time of 300 s, has been encoded with multiple descriptions using a technique based on channel splitting and motion-compensated temporal interpolation of missing frames [45], with a total coding rate of about 1.8 Mbps, resulting in an average PSNR of 39.94 dB for central decoding and 35.20 dB for side decoding.

The MANET has been simulated using the *ns-2* discrete event simulator, which models quite accurately the 802.11 MAC layer and implements a shadow/multi-path fading model. Even though the PHY model of ns-2 is somewhat inaccurate, its wide availability makes it one of the most commonly used research tools in the field of mobile ad-hoc networking (see, for instance, [46], [47], and references therein).

Our implementation of ABCD consists in a modified version of the 802.11 MAC layer agent that supports our scheme of reliable broadcast and a routing agent that implements the application logic. The mobile nodes' interface parameters are based on the specifications of the ORiNOCO 11 b/g card [22], which has a nominal range of 25 m using a Ricean propagation model for moving nodes, a Two-Ray Ground model for static, a path loss exponent $\beta = 2.7$, and a shadowing deviation $\sigma_{\text{dB}} = 6.8$. The parameter λ in the minimisation problem (2) has been found experimentally by maximising the average quality (in terms of PSNR) of the decoded sequences, for a value of $\lambda = 1.4$. However, preliminary tests with several values of λ showed that the technique is quite robust with respect to the choice of this parameter. Also, for obvious implementational reasons, problem (2) cannot be solved testing all $k \in \mathbb{N}$; in practice, it is reasonable to assume that the optimal value of k must lie in a interval $[0, k_{\text{max}}]$, with k_{max} relatively small, since a time too long to gain access to the channel would result in the packet being dropped for lateness. In order to find a suitable value for k_{max}, we ran several simulations with very large values of k_{max}, in order to be sure that the optimum was not missed; we found that the optimal values for k always lay between 0 and twice the limit prescribed by the standard, $2 \times 7 = 14$; the following simulations are therefore run with $k_{\text{max}} = 14$. From our simulations, we observed that the values of k actually chosen by the optimisation algorithm depend on the number of frames affected by the current packet: Intra-frame packets are the most protected (the most common value for k is 12), while packets containing frames with no dependency are the less protected (the most common value for k is 1).



(a) (b)

Figure 4. Histogram of frame delay. The vertical bar marks the maximum delay for frame decoding in the case of conversational pattern: (a) delay of a generic frame; (b) average delay of frames at a generic node.

In these experiments, we compare the two versions of the protocol (*i.e.*, plain ABCD and ABCD with CoDiO extension) in a network with 100 nodes and a density of 40 nodes per neighbourhood. This is an extremely high density, chosen in order to appreciate the capability of the proposed framework to deal with very harsh conditions of the network (*e.g.*, a group of rescuer rushing towards the injured in a disaster, or a maniple of soldiers converging on a target). Tests performed at different densities showed that the lower the density is, the more similar the performances of the two protocols are, which was to be expected, as the proposed framework is specifically designed for high densities.

In Fig. 4(a), we compare the histogram of the frame delay for the two versions of the protocol, collected from all the nodes of the network. Note that any frame with a delay higher than 100 ms (vertical bar) would not abide to the conversational pattern, and is therefore dropped. We can observe that, in the reference version, more than one half of the frames are too late to be decoded (55 %), while in the proposed version only a light tail of the histogram (2.7 %) crosses the deadline. This means that, in the reference technique, one half of the received packets are dropped as useless, even though the channel resources for their transmission have been spent. This can be also observed in Fig. 4(b), where we compare the histogram of the average delay perceived by the nodes: about one half of the nodes have, in the reference version, an average delay higher than the 100 ms threshold. On the other hand, the CoDiO framework allows to exploit the channel more efficiently, and no node experiences an average delay higher than 70 ms. We also present, in Fig. 5, the cumulative distribution function of the *maximum* frame delay experienced by a generic node. We observe that, in the reference scheme, all the nodes experience at least once a very high delay (larger than 200 ms) because of congestion. On the contrary, when CoDiO is used, about 40 % of nodes never experiences a delay larger than 100 ms and only about one third of them experiences a maximum delay larger than 200 ms. However, even those nodes cannot have an average delay larger than 70 ms as shown in Fig. 4(b).

Being discarded because of lateness is of course not the only reason for a frame not to be decoded by a node. A frame could have not been scheduled for sending in the first place (which can happen only in the proposed extension), or have exhausted all the requests granted without obtaining the channel (in both versions). This results in the fact that for some frames a node receives both, or one, or none of the descriptions. The reference technique uses central decoding for 73 % of frames and side decoding for 19 %, while the proposed technique uses central decoding 94 % of frames and side decoding for 5 % (concealment is used for the remaining frames). This result is mostly network-related and almost completely independent on the MDC scheme used (apart from the inter-frame dependency due to the predictive structure, see the definition of $\mathbf{I}$ in Sec. III), since using a particular MDC technique affects only the length of video packets, with a negligible effect on contention over channel access.

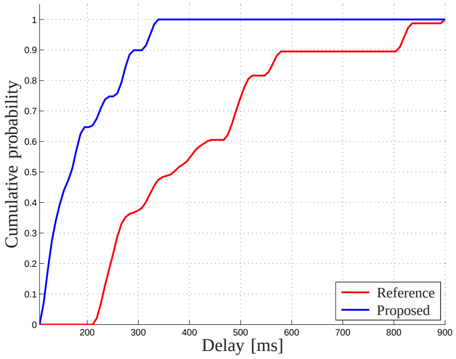
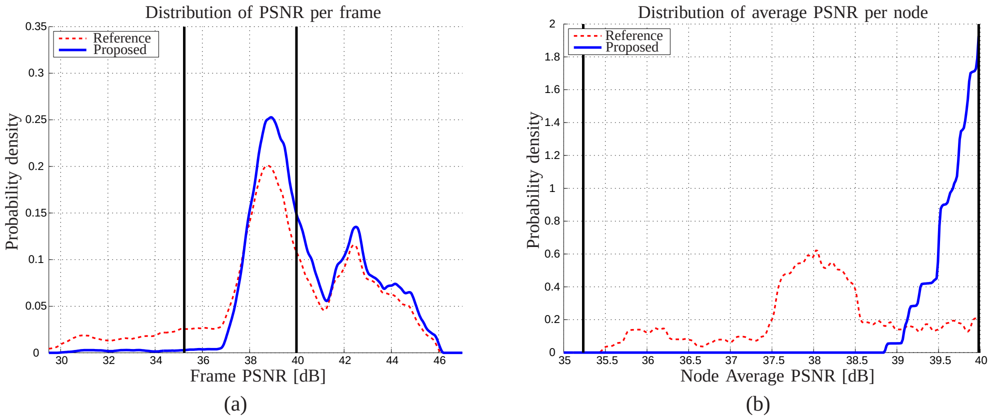


Figure 5. Cumulative distribution of maximum delay experienced by a generic node.

How the use of a decoding strategy reflects on the video quality depends on the codec used and the concealment strategy employed; in our experiments, we used an MDC technique that we introduced in [7], employing frame freezing as concealment. In Fig. 6(a) we compare, for the two versions of the protocol, the probability density functions of PSNR per frame, *i.e.*, considering the PSNR of each frame decoded by each node as a realisation; the PDF is then estimated with the Parzen window method [48]. Both distributions have roughly the same shape; however, in the reference scheme,



(a)

(b)

Figure 6. Probability density functions of the PSNR. The vertical bars mark the average video PSNR for the sequence when decoded with side (leftmost) and central (rightmost) decoder: (a) distribution of frame PSNR; (b) distribution of sequence PSNR.

nodes decode a generic frame with a low PSNR with a higher probability, for an average of 38.2 dB, while in the proposed technique frames are much more likely to be decoded with a high PSNR, for an average of 39.7 dB, *i.e.*, 0.24 dB short of the average central quality, clearly better than the reference. This means that, using the reference technique in this configuration, the effects of unreliable transmission are visible to the user, whereas in the proposed technique they are barely perceptible.

Now we consider, for each single node, the average PSNR of the locally decoded video. In Fig. 6(b) we show the estimated probability density function of the per-node PSNR; here, we see how the reference technique corresponds to an almost flat distribution, bounded between the average quality of side and central decoding. This implies that the video quality is acceptable for all nodes, but some of them have an average PSNR much smaller then others. On the other hand, the proposed technique has a much more peaked distribution, meaning that all nodes achieve a very high quality, with the modal value corresponding to the maximum quality. In other words, even though some frames are decoded with a relatively small PSNR (as shown in Fig. 6(a)), this hardly happens repeatedly to the same nodes; as a consequence, almost all nodes have an average PSNR over the sequence that is close to the maximum. The difference in dispersion can be better quantified by the inter-quartile range: in the reference technique the 25-th node, ordered by decreasing PSNR, has an average PSNR almost 1.0 dB higher than the 75-th, whereas in our proposed technique this range is in the order of 0.3 dB.

Furthermore, in order to test the performance of the protocol in presence of node mobility, two different models have been experimented: Random Way-point and Reference Point Group Mobility [45]. The delivery rate as a function of the nodes' average speed for the Random Way-point model (the most challenging for a structured protocol, as the nodes move independently) is presented in Fig. 7. We observe that the performance of the protocol is practically unaffected for speeds within the targeted application range (0–5 m/s), and starts to deteriorate only for speeds greater than 9 m/s.

In summary, we find a significant gain both in terms of PSNR and in average end-to-end delay, while the delivery rate is kept close to 100 %, making the technique suited for conversational video applications over mobile ad-hoc networks. These results have been obtained in experimental conditions of high bitrate, high density, random mobility, and large number of nodes, *i.e.*, conditions prone to generate a severe congestion on the channel; also, a stringent constraint on delay has been imposed. Tests have been performed in less harsh scenario as well, but – even though the proposed technique is never outperformed by the reference technique – the gain is less and less significant as congestion is less relevant (because a longer delay is accepted) or less likely to occur (because the node density and the bitrate are small); this of course depends on the fact that this framework is designed to grant conversational delivery in congested networks, and is unnecessary in more tolerant and less crowded networks.

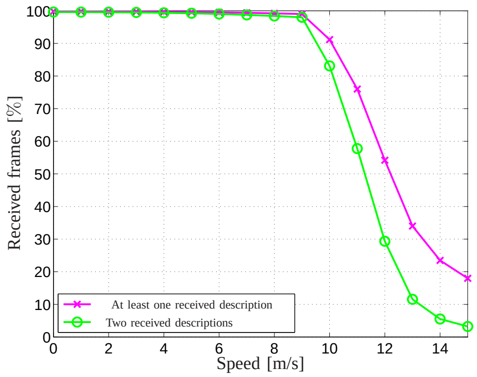


Figure 7. Case of mobility with Random Way-point: ABCD delivery rate as a function of average node speed.

A last remark is due about the overhead of the proposed protocol. We observe that the CoDiO framework does not introduce new packets with respect to the reference ABCD protocol, as all optimisation-related information is piggybacked either in video data or in attachment messages. In its turn, the message overhead of ABCD is detailed in our previous paper [22], and even in the worst scenario never surpasses 10 %. As far as the computational overhead is concerned, it is by far negligible with respect to the computational load required by the video decoding: it could be easily shown that the computation of $J(k)$ for all different k values can be performed with a few hundred clock cycles. Since the packet period is of the order of 10^{-1} s, CoDiO can be implemented with a few thousands cycles per second. For comparison, H.264/AVC baseline decoding requires several millions of cycles per second [49], even at as low resolutions as CIF, while multiple description decoding is even more complex.

V. Conclusions and future work

In this work, we proposed a congestion avoidance framework designed for multi-tree overlays in mobile ad-hoc environments with high node density and stringent constraints on delay. Namely, we introduced a dynamic adjustment of the MAC layer transmission parameters, optimised w.r.t. a congestion-distortion criterion. The model for congestion and distortion takes into account the video coding structure as well as the topology of the overlay network. In particular, for distortion estimation, we introduced an efficient way to propagate information about possible paths alternative to the multi-tree overlay, in order to classify the nodes in groups differently affected by the loss of a packet. The total distortion is then estimated by weighting the expected distortions of each groups with the estimated number of receiving nodes in each group. This allows to make a reliable prediction on the consequences of sending a packet with a particular retry limit, thus optimising video transmission in a CoDiO sense. As a result, both congestion and distortion are optimised in dense networks imposing a conversational pattern on the video stream.

Future work includes the introduction in the model of an up-link constraint, such that a node cannot stream more than one description at once. Such a constraint is justified both by the limited capacity of some mobile terminals, and by the high battery consumption associated with transmitting at a high bitrate. Some preliminary promising results have been obtained with a combination of MDC and network coding over an ABCD overlay [50], by allowing the active nodes to stream a combination of descriptions, rather than more than one description. We are also currently investigating some solutions integrating a rate-distortion optimised packet scheduling operated independently by each active node.

References

[1] M. Frodigh, P. Johansson, and P. Larsson, “Wireless ad-hoc networking: the art of networking without a network,” *Ericsson Review*, vol. 4, pp. 248–263, 2000.

[2] L. Hanzo and R. Tafazolli, “Admission control schemes for 802.11-based multi-hop mobile ad-hoc networks: a survey,” *IEEE Commun. Surveys Tuts.*, vol. 11, no. 4, pp. 78–108, Oct. 2009.

[3] M. Gerla, “From battlefields to urban grids: new research challenges in ad-hoc wireless networks,” *Elsevier J. on Pervasive and Mobile Comput.*, vol. 1, no. 1, pp. 77–93, 2005.

[4] V. K. Goyal, “Multiple description coding: compression meets the network,” *IEEE Signal Proc. Mag.*, vol. 18, no. 5, pp. 74–93, 2001.

[5] Y. Wang, A. Reibman, and S. Lin, “Multiple description coding for video delivery,” *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, 2005, invited Paper.

[6] J. G. Apostolopoulos, “Reliable video communication over lossy packet networks using multiple state encoding and path diversity,” in *Proc. of SPIE Visual Comm. and Image Proc.*, 2001.

[7] C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, “H.264-based multiple description coding using motion compensated temporal interpolation,” in *Proc. of IEEE Multimedia Sign. Proc.*, Saint-Malo, France, Oct. 2010.

[8] D. O. Reudink, “The channel splitting problem with interpolative coders,” Bell Labs, Tech. Rep., Oct. 1980, tM80-134-1.

[9] V. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 821–834, 1993.

[10] R. Puri and K. Ramchandran, “Multiple description source coding through forward error correction codes,” in *Proc. of Asilomar Conf. on Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, Oct. 1999, pp. 342–346.

[11] V. K. Goyal and J. Kovačević, “Generalized multiple description coding with correlating transforms,” *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2199–2224, Sep. 2001.

[12] C. Tillier, C. T. Petrişor, and B. Pesquet-Popescu, “A motion-compensated overcomplete temporal decomposition for multiple description scalable video coding,” *EURASIP J. Image Video Proc.*, vol. 1, pp. 1–12, 2007.

[13] I. Radulovic, P. Frossard, Y.-K. Wang, M. Hannuksela, and A. Hallapuro, “Multiple description video coding with H.264/AVC redundant pictures,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 144–148, 2010.

[14] J. Villalón, P. Cuenca, L. Orozco-Barbosa, Y. Seok, and T. Turletti, “Cross-layer architecture for adaptive video multicast streaming over multirate wireless LANs,” *IEEE J. Select. Areas Commun.*, vol. 25, no. 4, pp. 699–711, 2007.

[15] J. Monteiro, C. Calafate, and M. Nunes, “Robust multipoint and multi-layered transmission of H.264/SVC with raptor codes,” *Telecommunication Systems*, vol. 49, pp. 113–128, Jun. 2010.

[16] P. Chou, Y. Wu, and K. Jain, “Practical network coding,” in *Proc. of Allerton Conf. on Commun. Control and Comput.*, Monticello, IL, 2003.

[17] E. Magli and P. Frossard, “An overview of network coding for multimedia streaming,” in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, New York City, NY, USA, Jul. 2009, pp. 1488–1491.

[18] I. Stoica, R. Morris, D. Liben-nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup service for Internet applications,” *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 149–160, 2003.

[19] “Gnutella p2p network,” Website. [Online]. Available: http://rakjar.de/gnufu/index.php/Main_Page

[20] D. N. da Hora, D. F. Macedo, L. B. Oliveira, I. G. Siqueira, A. A. F. Loureiro, J. M. Nogueira, and G. Pujolle, “Enhancing peer-to-peer content discovery techniques over mobile ad-hoc networks,” *Elsevier J. on Computer Commun.*, vol. 1, pp. 1–15, 2009.

[21] M. Van der Schaar and N. Sai Shankar, “Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 50–58, 2005.

[22] C. Greco and M. Cagnazzo, “A cross-layer protocol for cooperative content delivery over mobile ad-hoc networks,” *Inderscience IJCNDS.*, vol. 7, no. 1–2, pp. 49–63, 2011.

[23] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[24] O. Badarneh and M. Kadoch, “Multiple description video multicast in MANETs,” in *Proc. of IEEE Int. Performance Comput. and Commun. Conf.*, 2010, pp. 222–230.

[25] P. Goudarzi and M. Hosseinpour, “Video transmission over MANETs with enhanced quality of experience,” *IEEE Trans. Consumer Electron.*, vol. 56, no. 4, pp. 2217–2225, 2010.

[26] E. Setton and B. Girod, “Congestion-distortion optimized scheduling of video over a bottleneck link,” in *Proc. of IEEE Multimedia Sign. Proc.*, 2004.

[27] X. Zhu, E. Setton, and B. Girod, “Congestion-distortion optimized video transmission over ad-hoc networks,” *Signal Proc. Image Comm. (Elsevier Science)*, vol. 20, pp. 1–21, 2005.

[28] VidTorrent, “http://viral.media.mit.edu/index.php?page=vidtorrent,” Website.

[29] S. Zezza, E. Magli, G. Olmo, and M. Grangetto, “Seacast: A protocol for peer-to-peer video streaming supporting multiple description coding,” in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, Jul. 2009, pp. 1586–1587.

[30] P. Cataldi, M. Grangetto, T. Tillo, E. Magli, and G. Olmo, “Sliding-window raptor codes for efficient scalable wireless video broadcasting with unequal loss protection,” *IEEE Trans. Image Proc.*, vol. 19, no. 6, pp. 1491–1503, Jun. 2010.

[31] C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, “ABCD : Un protocole cross-layer pour la diffusion vidéo dans des réseaux sans fil ad-hoc,” in *Proc. of GRETSI*, 2011.

[32] J. Tourrilhes, “Robust broadcast: improving the reliability of broadcast transmissions on CSMA/CA,” in *Proc. of IEEE Int. Symp. on Personal Indoor and Mobile Radio Commun.*, 1998.

[33] M. K. Marina, G. D. Kondylis, and U. C. Kozat, “RBRP: A robust broadcast reservation protocol for mobile ad-hoc networks,” in *Proc. of IEEE Int. Conf. on Commun.*, 2001.

[34] A. El Fawal, J. Le Boudec, and K. Salamatian, “Multi-hop broadcast from theory to reality: practical design for ad-hoc networks,” in *Proc. of IEEE/ACM Int. Conf. on Autonomic Comput. and Commun. Syst.*, 2007.

[35] E. Royer, P. Melliar-Smith, and L. Moser, “An analysis of the optimum node density for ad-hoc mobile networks,” in *Proc. of IEEE Int. Conf. on Commun.*, 2001.

[36] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, “Cross-layer design of ad-hoc networks for real-time video streaming,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 59–65, 2005.

[37] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin, “Multiple description coding in networks with congestion problem,” *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 891–902, Mar. 2001.

[38] J. Apostolopoulos, T. Wong, W.-T. Tan, and S. Wee, “On multiple description streaming with content delivery networks,” in *Proc. of IEEE Int. Conf. on Computer Commun.*, vol. 3, 2002, pp. 1736–1745.

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

[40] T. Maugey, T. André, B. Pesquet-Popescu, and J. Farah, “Analysis of error propagation due to frame losses in a distributed video coding system,” in *Proc. of Europ. Sign. Proc. Conf.*, 2008.

[41] T. Maugey and B. Pesquet-Popescu, “Side information estimation and new symmetric schemes for multi-view distributed video coding,” *Elsevier J. Vis. Comm. and Image Repres.*, vol. 19, no. 8, pp. 589–599, 2008.

[42] N. Tizon and B. Pesquet-Popescu, “Adaptive video streaming with a recursive distortion model,” in *Proc. of GRETSI*, Dijon, France, Sep. 2009.

[43] F. Ye, R. Yim, S. Roy, and J. Zhang, “Efficiency and reliability of one-hop broadcasting in vehicular ad-hoc networks,” *IEEE J. Select. Areas Commun.*, vol. 29, no. 1, pp. 151–160, 2011.

[44] B. Crow, I. Widjaja, L. Kim, and P. Sakai, “IEEE 802.11 wireless local area networks,” *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 116–126, 1997.

[45] C. Greco, G. Petrazzuoli, M. Cagnazzo, and B. Pesquet-Popescu, “An MDC-based video streaming architecture for mobile networks,” in *Proc. of IEEE Multimedia Sign. Proc.*, Hangzhou, PRC, Oct. 2011.

[46] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad-hoc network research,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, Aug. 2002.

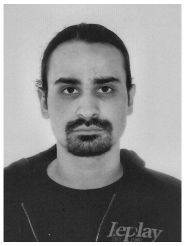
[47] C. Lochert, B. Scheuermann, and M. Mauve, “A survey on congestion control for mobile ad-hoc networks,” *Wireless Communications and Mobile Computing*, vol. 7, no. 5, pp. 655–676, 2007.

[48] E. Parzen, “On estimation of a probability density function and mode,” *Ann. of Math. Stat.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[49] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, “H.264/AVC baseline profile decoder complexity analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, Jul. 2003.

[50] I. Nemoianu, C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, “A framework for joint multiple description

coding and network coding over wireless ad-hoc networks,” in *Proc. of IEEE Int. Conf. Acoust., Speech and Sign. Proc.*, Kyoto, Japan, Mar. 2012.



Claudio Greco (S’08–M’12) received his B.Eng. in 2004 and his M.Sc. (with honors) in 2007, from the “Federico II” University of Naples, Italy. He is currently pursuing his Ph.D. degree in the Department of Signal and Image Processing of Télécom ParisTech, Paris, France.

His research interests include multiple description video coding, mobile ad-hoc networking, cooperative multimedia streaming, and network coding.

Marco Cagnazzo (M’05-SM’11) obtained the Laurea (equivalent to the M.S.) degree in Telecommunication Engineering from Federico II University, Napoli, Italy, in 2002, and the Ph.D. degree in Information and Communication Technology from Federico II University and the University of Nice-Sophia Antipolis, Nice, France in 2005.

He was a post-doc fellow at I3S Laboratory (Sophia Antipolis, France) from 2006 to 2008. Since February 2008 he has been Maître de Conférences (roughly equivalent to Associate Professor) at Institut Mines-TELECOM, TELECOM ParisTech (Paris), within the Multimedia team. He is author of more than 60 contributions in peer-reviewed journals and conferences proceedings. His research interests are region-based and content-adapted image coding, low complexity, scalable, robust, and distributed video coding, multiple description coding, video delivery over MANETs.

Dr. Cagnazzo is the Area Editor for *Elsevier Signal Processing: Image Communication* and *Elsevier Signal Processing*. Moreover he is a reviewer for major international scientific reviews (IEEE Trans. Multimedia, IEEE Trans. Image Processing, IEEE Trans. Signal Processing, IEEE Trans. Circ. Syst. Video Tech., *Elsevier Signal Processing*, *Elsevier Sig. Proc. Image Comm.*, and others) and conferences.

Béatrice Pesquet-Popescu (SM’06) received the engineering degree in telecommunications from the ”Politehnica” Institute in Bucharest in 1995 (highest honours) and the Ph.D. degree from the Ecole Normale Supérieure de Cachan in 1998.

In 1998, she was a Research and Teaching Assistant with Université Paris XI, Paris. In 1999, she joined Philips Research France, Suresnes, France, where she worked for two years as a Research Scientist, then as a Project Leader, in scalable video coding. Since Oct. 2000 she is with Télécom ParisTech (formerly, ENST), first as an Associate Professor, and since 2007 as a Professor, Head of the Multimedia Group. She is the Head of the UBIMEDIA common research laboratory between Alcatel-Lucent and Institut Télécom. Her current research interests are in source coding, scalable, robust and distributed video compression and sparse representations.

Dr. Pesquet-Popescu was an EURASIP BoG member (2003-2010), and an IEEE Signal Processing Society IVMSP TC member and MMSP TC associate member. She serves as an Associate Editor for IEEE Trans. on Image Processing, IEEE Trans. on Multimedia, IEEE Trans. on CSVT, Elsevier Image Communication, and Hindawi Int. J. Digital Multimedia Broadcasting journals and was till 2010 an Associate Editor for Elsevier Signal Processing. She was a Technical Co-Chair for the PCS2004 conference, and General Co-Chair for IEEE SPS MMSP2010, EUSIPCO 2012, and IEEE SPS ICIP 2014 conferences. Beatrice Pesquet-Popescu is a recipient of the ”Best Student Paper Award” in the IEEE Signal Processing Workshop on Higher-Order Statistics in 1997, of the Bronze Inventor Medal from Philips Research and in 1998 she received a ”Young Investigator Award” granted by the French Physical Society. She holds 23 patents in wavelet-based video coding and has authored more than 250 book chapters, journal and conference papers in the field. In 2006, she was the recipient, together with D. Turaga and M. van der Schaar, of the IEEE Trans. on Circuits and Systems for Video Technology ”Best Paper Award”.