Computer Networks 103 (2016) 84–100

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# Network coding-based approach for efficient video streaming over MANET☆

Olfa Ben Rhaiem[a,*], Lamia Chaari Fourati[a], Wessam Ajib[b]

[a] National school of engineering, LETI Laboratory, Sfax, Tunisia
[b] Department of Computer Science, Université du Québec à Montréal, Canada

ARTICLE INFO

*Article history:*
Received 26 September 2015
Revised 5 March 2016
Accepted 2 April 2016
Available online 9 April 2016

*Keywords:*
Video streaming
Cross-layer
MANET
Multicast
Network coding

ABSTRACT

Video streaming over mobile ad-hoc networks is becoming a highly important application for reliably delivering the content between the user and the content storage node. The key challenge is, hence, to address the impact of the user mobility on the quality of the delivered video. Accordingly, the pioneering concept of network coding (NC) emerges as a promising approach for improving the video transmission quality mainly in multicast mobile environment. This work focuses on improving the Quality of Service of video streaming over mobile ad-hoc networks using random NC. We consider video coded by the widely-used H264/SVC codec that generates packets with different priorities and provides traffic differentiation using the IEEE 802.11e MAC. Intuitively, focusing on lowering the error transmission of high priority packets leads to enhance the video streaming quality. Accordingly, this work develops and proposes a new scalable transmission scheme that decreases the loss of high priority packets. Our approach, named Extended Multicast Scalable Video Transmission using Classification-Scheduling Algorithms and Network Coding over MANET (and denoted EMSCNC), adopts a cross layer design between the H.264/SVC codec, the network and MAC layers. Moreover, we develop an analytical framework allowing the performance evaluation in terms of throughput, delay, and packet delivery ratio. Simulation results validate our analytical model and confirm the substantial performance improvement brought by our approach.



## 1. Introduction

Mobile ad-hoc networks (MANET) are characterized by unstable network topology. Also, wireless communication between MANET nodes has a broadcasting nature which may produce redundant packets and create a broadcast storm problem. Consequently, new approaches based on network coding (NC) have been proposed to address these problems. NC combines many packets in one coded packets instead of using the typical forwarding techniques of store and forward. It greatly improves the transmission performance by reducing the number of packet transmissions. Hence, it provides less delivery delay and higher data throughput.

NC was introduced in [6,7] for wired multicasting transmission where it was shown that it may achieve higher throughput than traditional routing solutions by linearly combining data packets at intermediate nodes. Later, linear network coding scheme [8,9] and random network coding [10,11] have been investigated to improve the transmission performance in different networking scenarios. The challenges resulted from bandwidth constraint and the dynamic topology of MANET makes the multicast and multi-hop supported routing hot research topics. Additionally, scalable video coding (SVC) approach is applied to enable efficient video streaming. H.264/SVC [12] is widely-used to effectively compress the video. It provides temporal, spatial and quality scalabilities. H.264/SVC packets have different priorities. High-priority packets are the most important for guaranteeing high video transmission quality.

A random network coding for MANET named CodeCast is proposed in [13]. It is able to achieve high throughput with low overhead and low latency. However, CodeCast has its own limitations, such as assuming constant bit-rate video, equal-size packets, etc. Also, it does not increase the probability of recovering high priority packets but provides the same loss probability for all the video packets regardless of their priority. Authors in [14] aimed at improving Codecast by proposing a new network coding based scheme (E-Codecast) to maximize the overall video quality at all destinations under the constraint of network capacity. In fact they proposed a low-complexity optimization algorithm to optimize the packet forwarding frequency and to support scalable video coding.

☆ This work is published in the proceedings of IEEE VTC–fall 2015 [1].

* Corresponding author. Tel.: +21626153192.
*E-mail addresses:* olfa.benrhaiem@gmail.com (O. Ben Rhaiem), lamia.chaari@enis.rnu.tn (L. Chaari Fourati), ajib.wessam@uqam.ca (W. Ajib).

http://dx.doi.org/10.1016/j.comnet.2016.04.002
1389-1286/

By simulation, authors show that their scheme is capable of increasing the network resource efficiency and video quality.

A network coding-based real-time multicast (NCRM) protocol is proposed in [15] for MANET to reduce the energy consumption. It combines the PUMA (protocol for unified multicasting through announcement) approach [16] with random linear network coding (RLNC). PUMA allows eliminating the unnecessary packets in order to reduce the network overhead and the end-to-end-delay. NCRM can both reduce the energy consumption and increase the throughput. Moreover, it is shown to be robust in high mobility and high density environment. However, this approach proposes to simultaneously send the whole block of coded packets which may lead to large delays, serious network congestions and low packet delivery ratios.

To address the low packet delivery ratio and low delay, a new scheme (named PNCRM) was proposed in [17] combining partial network coding [18] and the real-time PUMA. PNCRM is based on RLNC but each vector of packets is transmitted partially. It significantly increases the reliability and throughput. Anyhow, it performs poorly in multicasting scenarios with high traffic load.

The state-of-the-art survey shows that several approaches allow implementing NC in MANET. Since it is difficult to frequently update the network topology, mainly in high-mobility environment, RLNC is more suitable for ensuring high throughput and low network load. Although NC improves the throughput and the loss ratio, it does not guarantee to recover high priority packets. High-priority packet (considering H.264/SVC codec) loss is a basic QoS parameter for video coding in MANET. Additionally, some packets could be lost so that the Global Coefficient Matrix (GCM) [15] associated with RLNC cannot have the full rank for inversion to decode the packets at the receiver nodes.

The previously mentioned arguments motivate us to focus on minimizing the high priority packet loss for real time traffic in MANET. The main contributions of this paper are summarized as follows:

- We propose a new scheme named Extended Multicast Scalable Video Transmission using Classification Scheduling Algorithms and Network Coding over MANET (and denoted EMSCNC). It is an extension of our previous schemes named MSVT_CSA_NC [1].
- The proposed scheme adopts a cross layer design, achieves high performance and outperforms the state-of-the-art algorithms.
- This paper provides analytical models to estimate the average end-to-end delay, throughput and packet delivery ratio of EMSCNC.
- The proposed models are validated using numerical analysis as well as simulations.

The main characteristics of the proposed algorithm are:

- It integrates the H.264 scalable video coding (SVC), which is a more promising strategy than constant bit-rate video coding scheme.
- It minimizes the high priority packet loss by guaranteeing a successful reception of packets from the base layer (I or P frames). The authors in [3] describes thoroughly the H.264/SVC standard.
- It proposes a network coding scheme that, for the first time, provides the same block size for all the packets of one GOP.
- It creates coded packets based on their priorities by using an inter-layer compensation algorithm.

Our proposed scheme, EMSCNC, requires two workflows associated respectively to the source-node and intermediate (relay) node. The workflow on the source-node provide two algorithms: i) a classification algorithm that adjusts the *block_size* parameter of RLNC based on an inter-layer compensation; ii) and encoding algorithm to form blocks as a function of packets priorities. The second workflow, associated to the intermediate node, focuses on improving the decoding process.

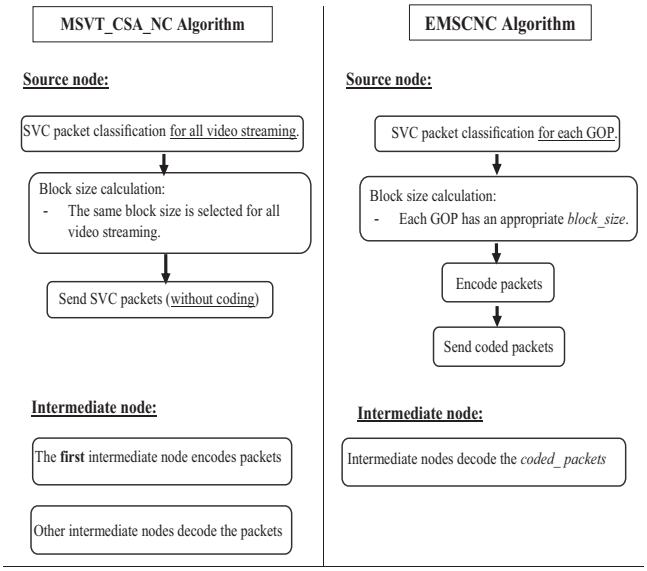



**Fig. 1.** MSVT_CSA_NC Vs EMSCNC.

The remainder of this paper is organized as follows. Section 2, illustrates the important difference between EMSCNC and the previous approach MSVT_CSA_NC. Section 3 summarizes the network model used to analyze the new scheme. The proposed transmission scheme is outlined in Section 4. In Section 5, simulation results validates our analytical results in Section 6. Finally, section 7 draws the concluding remarks.

## 2. MSVT_CSA_NC Vs EMSCNC

Our previous proposed approach named Multicast Scalable Video Transmission using Classification-Scheduling Algorithms and Network Coding over MANET (and denoted MSVT_CSA_NC), adopts also a cross layer solution between H.264/SVC codec, network and MAC layers. The difference between MSVT_CSA_NC and EMSCNC solution are at the source and intermediate nodes processing. For MSVT_CSA_NC, the source node performs a classification algorithm to adjust the block_size parameter of RLNC. The intermediate-node focuses on enhancing the RLNC and making it able to form blocks according to packets priorities. On the other hand for EMSCNC, the source node performs (i) packet classification and dynamic bloc size calculation based on inter-layer compensation, and ii) packet encoding process. These two functionalities are performed for each GOP. The intermediate node decodes the coded packet generated by the source node and re-encodes it before forwarding it. Comparing both solutions, we can also highlight that:

- EMSCNC provides the same block size for the packets of one GOP, contrarily to MSVT_CSA_NC where all the svc packets have the same size. Hence EMSCNC is able to guaranty an efficient coding that ensure a minimum loss of high priority packet.
- The loss of one packet in MSVT_CSA_NC increases the number of out-of-block packets whereas in EMSCNC it increases the same number for the same GOP only.
- EMSCNC reduces the end-to-end delay compared to MSVT_CSA_NC.

A comparison summery of the two schemes is depicted in Fig. 1:

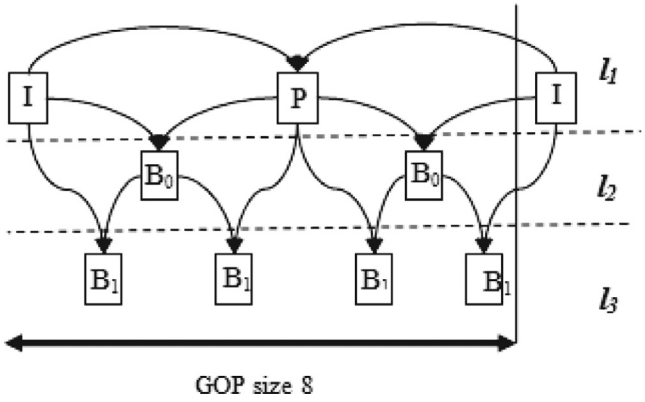



**Fig. 2.** SVC coding scheme for three layers with I-frames, P-frames and B-frames.

## 3. System model

The considered system consists of one single sender, denoted *Src*, that aims to multicast a set of video packets $p_1, p_2, \ldots, p_N$ to a set of receivers, denoted $\Omega$. Each receiver is connected to the sender via a wireless channel with channel rate $\mu$. The time is slotted and the slots are denoted by $t = 1, 2, \ldots$. The sender node transmits one coded packet at each time slot.

We consider that the layered video data is chunked where each chunk corresponds to a fixed number of frames that we refer to as a group of pictures GOP. Let $G$ denote the total number of GOP and $g_j$ denote the number of packets belonging to GOP $j \in \{1, \ldots, G\}$. Each GOP is encoded into $L$ layers. Let $N$ denote the total number of SVC generated packets and $l_i$ is the level of priority $l_i = l_1, l_2, \ldots, l_L$. We consider that a level $l_i$ for each GOP is composed of $N_i^j$ SVC packets, where $i \in \{1, \ldots, L\}$ defines the $L$ levels and $j \in \{1, \ldots, G\}$. Thus, in the intermediate node, $g_j$ and $N$ are given by:

$$g_j = \sum_{i=1}^{L} (N_i^j); \quad j \in \{1, \ldots, G\} \tag{1}$$

$$N = \sum_{j=1}^{G} (g_j) = \sum_{j=1}^{G} (\sum_{i=1}^{L} (N_i^j)) \tag{2}$$

Recall that although the number of frames per each GOP is fixed, $g_j$ and $N_i^j$ take different values for different GOP. To explain the coding scheme considered in this work, Fig. 2 shows an example with three SVC layers that support temporal scalability. This latter is used to increase the frame quality by increasing the frame display rate. It is achieved by three kinds of frames: I-frames, P-frames and B-frames. P-frames can only be decoded with the previous I or P frames, whereas, B-frames can be decoded using the next or the previous frames. Dislike the base layer (BL), the enhancement layer (EL) cannot be decoded by itself. Moreover, more enhancement layer packets are decoded, higher video quality is achieved. Thus, guaranteeing a successful reception of packets from the base layer (I and P) is the focus of this work.

Some notations used in this work are given in Table 1.

**Table 1**
Notations.

| Notation | Description |
|---|---|
| $N$ | Total number of SVC packets |
| $l_i$ | The $i$th level |
| $g_i$ | Number of packets belonging to the $i$th GOP |
| $G$ | Number of GOP |
| $\tau$ | Average throughput |
| $\mu$ | The bit rate |
| $q_i^j$ | The $i$th queue of the $j$th GOP |
| $Src$ | The source node |
| $R_i$ | The $i$th receiver |
| $r(n)$ | The transmission range |
| $n_h$ | Number of hops from $Src$ to $R_i$ |
| $D_{E2E}(R_i)$ | Average end to end delay for the $i$th receiver |
| $e(t)$ | Time taken for each block |
| $E_T(b, G)$ | The expected delay for $b$ blocks |
| $r_i^j$ | The $i$th number of out-of-block packets for the $j$th GOP |
| $PDR$ | Packet delivery ratio |
| $T_{slot}$ | Duration of one time slot |
| $d_c^i(b_{sz}^j)$ | Time required to code the $i$th block for a given $b_{sz}^j$ |
| $d_d^i(b_{sz}^j)$ | Time required to decode the $i$th block for a given $b_{sz}^j$ |
| $AIFS$ | Arbitration interframe space |
| $AIFSN$ | The AC-specific AIFS number |
| $SIFS$ | Length of short inter-frame space |
| $(x_i \times y_i)$ | The area size |
| $E_{Dist}(x_i, y_i)$ | Distance between $Src$ and $R_i$ on an area of size $(x_i \times y_i)$ |
| $CW$ | Contention window |
| $\delta$ | Number of receivers |
| $N(\delta, t)$ | Total number of packets set by the source and successfully received by all receivers |
| $T_s$ | Duration of each successful received packet. |
| $\Omega$ | Set of receivers |

## 4. Proposed transmission scheme

The new proposed transmission scheme aims to improve the efficiency of video streaming measured by the average delay denoted by $D_{E2E}$, average throughput, average PDR (Packet Delivery rate), average PSNR (Peak Signal to noise ratio) and jitter. First we propose a dynamic block size estimation to protect BL packets (level $l_1$) from being lost. Then, we present the proposed modification on random NC. In fact, the proposed EMSCNC scheme provides a cross-layer interaction among the application, network and MAC layers. The network coding algorithm operates at the network layer.

### 4.1. Source node

Two functionalities are provided by the source node, namely (i) packet classification and dynamic bloc size calculation based on inter-layer compensation, and (ii) packet encoding process.

#### 4.1.1. SVC packet classification and bloc size calculation

This work assumes that the application layer generates SVC packets of equal size. In traditional RLNC, the intermediate node is responsible of receiving native packets from the source node and grouping them into different blocks, each of $b_{sz}$ packets. Packets belonging to the same block may have different priorities. Also, in traditional RLNC a high number of out-of-block packets may exist. We denote $r_i^j$ as the number of out-of-block packets of the $i$th layer at the $j$th GOP; where $i \in \{1, 2, \ldots, L\}$, $j \in \{1, \ldots, G\}$. The priority of $r_i^j$ packets may be high. In this case, some high priority packets are lost, which decreases the transmission quality. The main objective of the proposed modifications to RLNC is to make it able to form blocks according to the priorities of the packets. Hence, packets belonging to the same block will have the same priority.

We assume that only packets belonging to the same GOP can be encoded together. Hence, the $j$th GOP has $N_{BL}^j$ packets from the base layer and $N_{EL}^j$ from the enhancement layers. Then the total number of packets belonging to the $j$th GOP is expressed by $g_j = N_{BL}^j + N_{EL}^j$; where $j \in \{1, \ldots, G\}$. In this work, the main concern before encoding packets is to choose for each GOP the appropriate size of blocks minimizing both $r_1^j$ packets of the BL having high priority and the total number of blocks.

At the network layer of the source, we propose to classify the packets, received from the upper layer (application layer), according to the value of (*DID, TID, QID*) triplet of identifiers. The


**Fig. 3.** SVC NALU header structure.

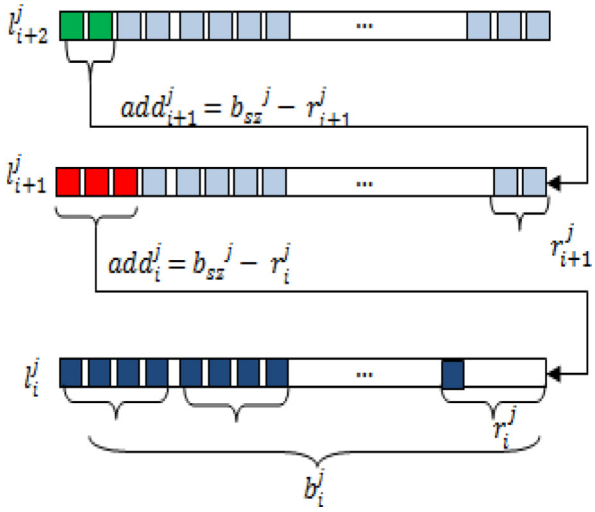



**Fig. 4.** Inter-layer block compensation mechanism with $b_{sz} = 4$, for the $j$th GOP.

dependency_id (DID), temporal_id (TID) and quality_id (QID) triplet for spatial, temporal and quality scalability respectively, indicates the priority level for each type of scalability. This information is located in the header structure of each NALU as shown in Fig. 3.

Hence, the priority of each packet can be easily identified. For example, with the temporal scalability, the TID field is coded on three bits, leading to $2^3$ queues having different priority levels (*i.e.*, $(0,1,0), (0,2,0), \ldots, (0,8,0)$). Packets belonging to the same queue $q^j_i$ have the same priority; $j \in [1,\ldots,G], i \in \{1,\ldots,L\}$. Next, we propose to calculate the appropriate block size (denoted $b_{sz}$). Thus, at the network layer, this value is added as a new field called:"*block_size*" in the header structure of the encoded packet. We assume that $b_{sz} \in [3,\ldots,12]$. Then, we choose the size of the $j$th GOP, (denoted $b^j_{sz}$) giving the minimum of $r^j_1$ packets which is defined by:

$$r^j_1 = min(b^j_{sz}) \tag{3}$$

If the same value of $r^j_1$ is obtained for different size values, the appropriate size that reduces the number of blocks for a given $g_j$ will be chosen. Indeed, the minimum number of blocks can be obtained by choosing the largest block size $b_{sz}$. For example, suppose that $N^j_1 = 23$, thereby $(23 \bmod 7 = 2)$ and $(23 \bmod 3 = 2)$; hence, $r^j_1 = 2$. In such case, the chosen $b_{sz}$ which reduces the number of blocks is 7.

As a result, the level $l^j_i$ has $b^j_i$ blocks; where $b^j_i = \frac{N^j_i}{b^j_{sz}}$. The size of the last formed block of each level $l^j_i$ may be less than the defined block size $b^j_{sz}$. As consequence, in each level $l^j_i$, these packets are considered lost since they can't form a block. In this situation, the probability $P_{loss}(l^j_i)$ of packet loss for the level $l^j_i$ is given by:

$$P_{loss}(l^j_i) = \frac{r^j_i}{[(b^j_i - 1) \times b^j_{sz} + r^j_i]}; \tag{4}$$

where $i \in \{1,2,\ldots,L\}$, and $j \in \{1,\ldots,G\}$

Since increasing the number of high priority coded packets guarantees better QoS, we propose a scheme that automatically forms an additional block for the different priority queues starting from the highest priority level queue which is $l^j_i$. The proposed

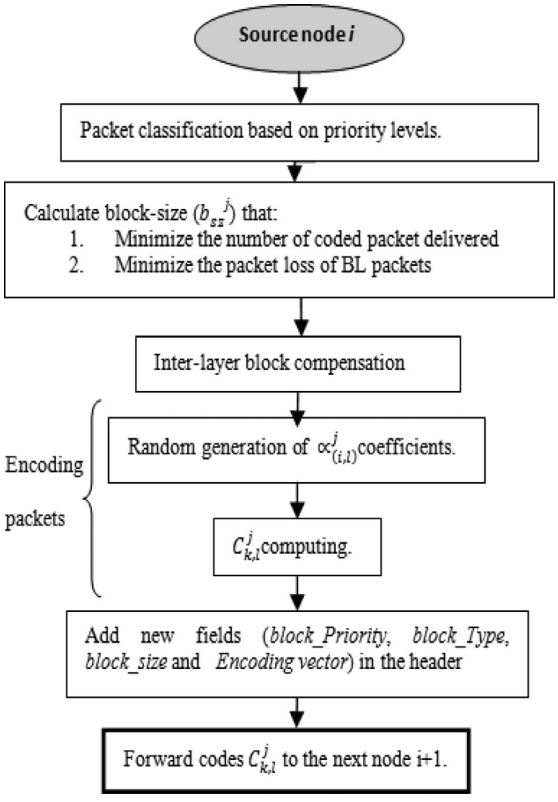



**Fig. 5.** Source node processing.

scheme assigns $add^j_i$ packets from the level $l^j_{i+1}$ to the queue $q^j_i$.

$$add^j_i = b^j_{sz} - r^j_i \tag{5}$$

As consequence, the number of out-of-block packets in queue $q^j_i$ becomes zero. Therefore, the probability of packet loss for the priority queue $q^j_1$ becomes zero and all $N^j_i$ packets are coded.

Fig. 4 illustrates our proposed priority-based packet loss reduction mechanism with $b^j_{sz} = 4$.

For example, for $N^1_{BL} = 17$, our algorithm chooses blocks of size 4 resulting into only 1 out-of-block packet (*i.e.*, $7 \bmod 4 = 1$). Therefore, three packets are required $(add_i = 3)$, from the level $i+1$ and added to the queue $i$ having the highest priority. By this way, it is guaranteed to have no non-encoded highest priority packets $P_{loss}(l_i) = 0$.

#### 4.1.2. Encoding packets

After selection the $j$th desired block size and before the coding, an additional arithmetic operation is performed. In fact, for each GOP $j$, when the block size is reached, this block of $b^j_{sz}$ packets will be multiplied by random coefficients $\alpha^j_{(l,i)}$, $l = 1,\ldots,L$; $i = 1,\ldots,b_{sz}$, and $j = 1,\ldots,G$ from the Galois field of size $2^F$ as shown in Eq. (6) (here F is the order of the Galois Field). The set of the coefficients for each level forms a GCM. Eq. (7) shows the GCM matrix for the level $l_1$.

$$C^j_{(k,l)} = \sum_{i=1}^{b_{sz}} \alpha^j_{(l,i)} \times P^j_{(l,i)} \tag{6}$$

where $l \in \{1,\ldots,L\}$, $j \in \{1,\ldots,G\}$ and $k = 1,\ldots,b^j_1$.

$$\begin{bmatrix} C^j_{(1,1)} \\ \vdots \\ C^j_{(b^j_1,1)} \end{bmatrix} = \begin{bmatrix} \alpha^j_{(1,1)} & \cdots & \alpha^j_{(b_{sz},1)} \\ \vdots & \ddots & \vdots \\ \alpha^j_{([(b^j_1-1)\cdot b_{sz}+1],1)} & \cdots & \alpha^j_{(b^j_1 \cdot b_s z,1)} \end{bmatrix} \begin{bmatrix} P^j_{i,1} \\ \vdots \\ P^j_{(b_{sz},1)} \end{bmatrix} \tag{7}$$

**Table 2**
Added new fields details.

| Field | Utility |
|---|---|
| **Block_priority** | **Get information from TID field located in NALU header.** |
| *Block_type* | Indicates the type of packets: "original_packet" or "coded_packet". |
| *Block_size* | Indicates the number of packets to be encoded for each GOP. |
| *Encoding_vector* | It is a random coefficient chosen from a finite field. |

**Table 3**
Mapping between Access categories (ACs) and *Block_priority* Field.

| Frame Type | TID (NALU) | Access categories |
|---|---|---|
| Frame I or P (from base layer: level 0) | 0 | AC0 (high priority). |
| Frame B1(from enhancement layer 1 (level 1) | 1 | AC1. |
| Frame B2 ( from enhancement layer 2) (level 2) | 2 | AC2. |
| Not used | | AC3 (useful for heterogeneous traffic). |

The coded packet format includes a header and data fields. Additional header fields are proposed to be added in the header structure of the encoded packet, called as *block_Priority*, *block_Type*, *block_size* and *Encoding_vector*.

In fact, since blocks are formed according to their priorities, a 2-bits field called *block_Priority* (this information can be obtained from the TID field in the SVC NALU header structure) is added. *Block_Priority* designs the priority level of each coded packet. Since a new packet type will be added (coded packet), we introduce a new field called *block_Type* set to "PT_Coded". When the coded packet is transmitted on the network it is accompanied by its *Encoding_vector*. The encapsulation of these fields occurs at the network layer. Anyway, the occurred additional time is negligible compared to the end-to-end delay transmission. In fact, our end to-end-delay transmission is measured in terms of second; whereas, the occurred additional time is given in terms of micro seconds. Hence, it is negligible compared to the end-to-end delay transmission. More details are presented in the following reference [4]. After that, this formed packet is forwarded. To improve the delay, new packets are transmitted while others are still in the coding process. The diagram given in Fig. 5 illustrates the source node processing.

#### 4.1.3. MAC layer: priority mapping

The IEEE 802.11e standard is designed to enhance the MAC mechanism by providing a QoS method called Enhanced Distributed Channel Access (EDCA). EDCA has four transmission queues: AC_BK (for background traffic), AC_BE (for best effort traffic), AC_VI (for video traffic) and AC_VO (for voice traffic). According to H.264/SVC, packets are assigned different priorities. Our work considers three priority levels: level 0 (packets from I and P images (BL), level 1 (frames from B1 images) and level 2 (packets from B2 images). Level 0 has the highest priority (Pr = 0). EDCA does not take in consideration the packet priority generated by the SVC coder (coded packet from the BL or for EL).

Thus, to increase the granularity of EDCA access categories we propose to allow prioritization between different coded packets (produced by NC) for the same video stream. Table 2 shows the NC fields of our proposed scheme. To indicate the correspondence between these NC fields and our proposed scheme we restrict our attention to *Block_priority* Field. Table 3 illustrates the mapping between Access categories (ACs) and *Block_priority* Field (used for network coding). It also illustrates the correspondence between the I, P and B frames and the 802.11e access categories (AC [0 to 3]). In fact, instead of attributing AC_VO AC_VI, AC_BE and AC_BK for voice, video, Best effort and background respectively, we propose to assign priority levels of the coded_packets (encoded in Block_Priority field) 0, 1 and 2 correspond to AC0, AC1 and AC2 respectively whereas AC3 can be useful for heterogeneous traffic. AC3 does not correspond to a priority level. (see Fig. 6).

### 4.2. Intermediate node (relay node)

At the network layer, once an intermediate node receives a coded packet, it de-encapsulates the packet and analyze its three header fields (*block_Priority*, *block_Type*, *block_size*). If the *block_Type* field interpreted by the intermediate node is *PT_Coded*, then, as shown in Eq. (8), the intermediate node can easily decode the received block of packets.

$$\begin{bmatrix} P^j_{l,1} \\ \vdots \\ P^j_{(b_{sz},1)} \end{bmatrix} = \begin{bmatrix} \alpha^j_{(1,1)} & \cdots & \alpha^j_{(b_{sz},1)} \\ \vdots & \ddots & \vdots \\ \alpha^j_{([(b^j_1-1)\cdot b_{sz}+1],1)} & \cdots & \alpha^j_{(b^j_1\cdot b_s z,1)} \end{bmatrix} \begin{bmatrix} C^j_{(1,1)} \\ \vdots \\ C^j_{(b^j_1,1)} \end{bmatrix} \quad (8)$$

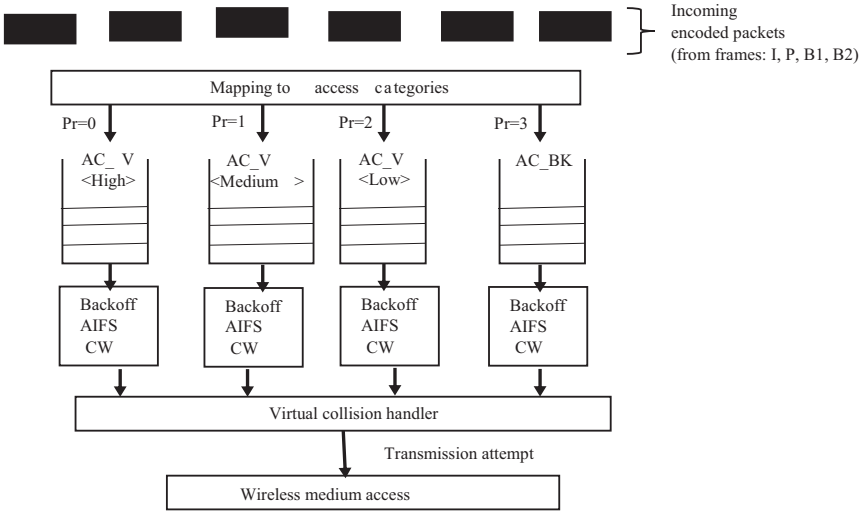



**Fig. 6.** Priority mapping (EDCA mechanism).

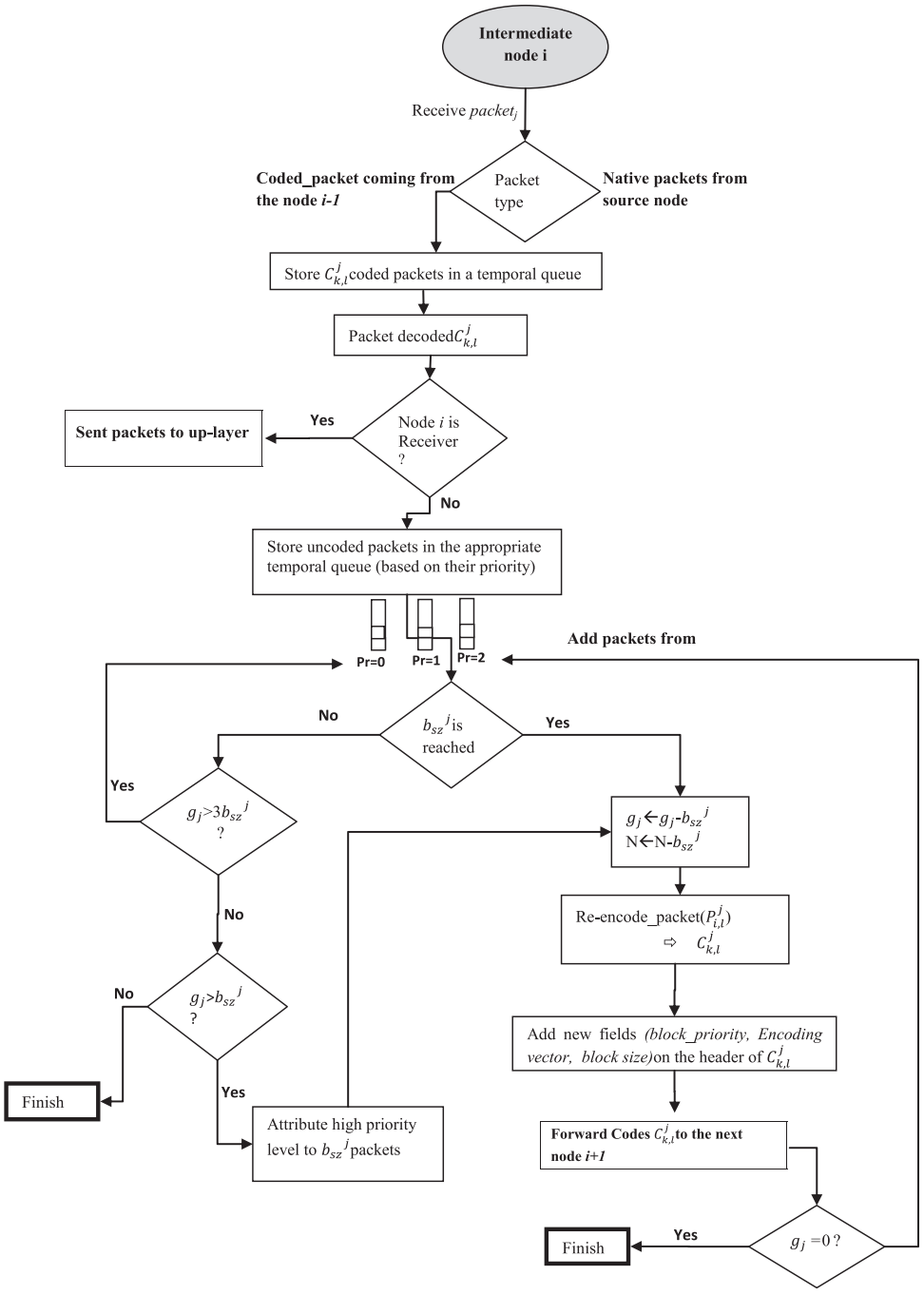



**Fig. 7.** different steps considered in this work at the intermediate node for each GOP.

For each GOP $j$, the intermediate node creates $k$ queues $(q_1^j, \ldots, q_k^j)$ where $k$ is the priority level and $q_1^j$ has the highest priority. In our case $k = 3$ since there are three quality levels according to the standard H.264/SVC. Each decoded packet is stored in the local memory of the appropriate queue to facilitate the re-encoding stage. The packets I or P with TID = 0 are stored in the first local memory with (Pr = 0). Packets B1 with TID = 1 are stored in the second local memory with (Pr = 1). Finally, packets B2 with TID = 2 are stored in the third local memory with (Pr = 2). It is worth noting that it is the same classification used in the priority mapping algorithm $AC_i$ and the coded_priority. Then, this "intermediate" node checks whether it is the receiver (final destination) or not. If this node is not a final destination, NC offers its great ability of re-encoding. In each queue, the number of received packets is calculated. Two possible cases can be distinguished at each given time. In the first case, the block size is reached, then a new combination of packets is generated. As a result, the number of delivered packets for each GOP and the total number of SVC packets decreases; $g_j = g_j - b_{sz}^j$ and $N = N - b_{sz}^j$ respectively. Then, the actual node $n_i$ transmits the re-encoded packets to the next node $n_{i+1}$. This process is repeated until $g_j = 0$. In the second case, if the block size is not reached and there are no new received coded packets. Therefore, each node computes the rest of packets (denoted as $remainedg_j$) kept in the queues and

compares it to $3 \times b_{sz}^{j}$. The $remained g_j$ is given by:

$$remained\, g_j = \sum_{k=1}^{3} p_{i,k}^{j};\ i = 1, 2, \ldots \tag{9}$$

where $p_{i,k}^{j}$ is the $i$th packet belonging to the $k$th queue.

If $g_j \leq 3 \times b_{sz}^{j}$, then this node checks if $g_j \geq b_{sz}^{j}$. If yes ($g_j \geq b_{sz}^{j}$), this intermediate node selects the RLNC coefficient and generates a new block. This block is assigned the highest priority among its packets. Then the $g_j - b_{sz}^{j}$ packets are discarded. If not, the process is finished.

Decoding and re-encoding process of the intermediate node are illustrated by Fig. 7.

Algorithm. 1 resumes the intermediate node processing.

**Algorithm 1:** Intermediate node processing.

**Input**: $\langle coded\_packet(C_{(k,l)}^{j})\rangle$

1 $l = 1, \ldots, 3$;
2 $j = 1, \ldots, G$;
3 $k = 1, \ldots, b_i^j$;
4 **if** $C_{k,l}^{j}.block\_Type$= *"PT_Coded"* **then**
5  $p_{(i,l)}^{j} = decode\_packets(C_{(k,l)}^{j})$;
6  **if** *node* $n_i$ = *"receiver"* **then**
7  sent packets to upper layer;
8  **else**
9  Create three temporal queues $q_l^j$;
10  $s_l^j = enqueue\_packets(p_{i,l}^{j})$;
11  **if** $s_l^j = b_{sz}^{j}$ **then**
12  $C_(k,l)^{j}$= $re - encode\_packets(p_{i,l}^{j}, b_{sz}^{j})$;
13  forward coded packets to node $n_{i+1}$;
14  $g_j = g_j - b_{sz}^{j}$;
15  $N = N - b_{sz}^{j}$;
16  **if** $g_j = 0$ **then**
17  Finish;
18  **else**
19  Return to line 10;
20  **else if** $g_j \geq 3b_{sz}^{j}$ **then**
21  Return to line 10
22  **else if** $g_j \geq b_{sz}^{j}$ **then**
23  Attribute high priority level to $b_{sz}^{j}$ packets;
24  Return to line 12;
25  **else**
26  $g_j \geq b_{sz}^{j}$ are discarded ;

## 5. Analytical model

We analyze the performance of our proposed scheme in terms of four metrics: average throughput, denoted $\tau$, average end-to-end delay, average PDR and average PSNR. The average throughput refers to the average rate at which data packet is delivered successfully from source to destination. It is usually measured by *bits*/s. Average end-to-end delay is the time taken for all data packets associated to a sequence video (300 frames) to reach its destination. Packet delivery ratio is the ratio of data packets successfully delivered to the destinations to those generated by the sources.

### 5.1. Average delay analysis (estimation)

Based on our network model, we here aim to model the average end to end delay (denoted $D_{E2E}$) between the source *Src* and destination nodes. Nodes move according to the random waypoint mobility model [19] over a system area of size ($x \times y$). Nodes have the same transmission range $r(n)$. Thus, two nodes can establish a wireless link only if they are within the range of each other (see Fig. 8). As in [5], we note by $n_h$ as the minimum number of hops between *Src* and one destination *R* over an area of size ($x \times y$). $E_T(b, G)$ denotes the average expected time, for one hop, taken by *G* GOP, grouped into *b* blocks.

We define by *d* the distance between two successive nodes $n_i$ and $n_{i+1}$. If the distance is set to $d(n_i, n_{i+1}) = r(n)$ where $r(n)$ is the transmission range of node *n*, then the end to end delay from *Src* to *R* can be obtained by:

$$D_{E2E}(R) = E_T(b, G) \times n_h \tag{10}$$

The minimum expected number of hops $n_h$ traversed by one block between $Src - R$ in multi-hop ad hoc networks on a rectangular area of size ($x \times y$) can be expressed as follow:

$$n_h = \frac{E_{Dist}(x, y)}{r(n)} \tag{11}$$

where the expected value of the distance, denoted by $E_{Dist}$ in a rectangular area is [1].

$$E_{Dist}(x, y) = \frac{1}{15}\left[\frac{x^3}{y^2} + \frac{y^3}{x^2} + \sqrt{x^2 + y^2}\left(3 - \frac{x^2}{y^2} - \frac{y^2}{x^2}\right)\right] + \frac{1}{6}\left[\frac{y^2}{x} arcosh\frac{\sqrt{x^2 + y^2}}{y} + \frac{x^2}{y} arcosh\frac{\sqrt{(x^2 + y^2)}}{x}\right] \tag{12}$$

where

$$arcosh(a) = \ln\left(a + \sqrt{a^2 - 1}\right)$$

We model the network topology as a geometric graph, where one *Src* transmits *N* packets to random receivers $R_i \in \Omega$. Let $\delta = card(\Omega)$ be the number of receivers. Thus, for each receiver $R_i$; $i \in [1, 2, \ldots, \delta]$, corresponds a new rectangular area of size ($x_i \times y_i$); where $x_i = \alpha_i \cdot x$ and $y_i = \beta_i \cdot y$, ($\beta_i$, $\alpha_i$) $\in$ ]0, 1[. For example, as shown in Fig. 8, the receiver $R_1$ forms a rectangular area of size ($\alpha_1 \cdot x \times \alpha_1 \cdot y$). Let $E_{Dist}(x_i, y_i)$ be the distance between *Src* and $R_i$ on an area of size ($x_i \times y_i$). The expected number of hops $n_h$ will be expressed as follow:

$$n_h = \frac{E_{Dist}(x_i, y_i)}{r(n)} \tag{13}$$

where, $x_i = \alpha_i \cdot x$ and $y_i = \alpha_i \cdot y$; $\beta_i$, $\alpha_i \in$ ]0, 1[

Therefore, according to Eq. (10) the average delay for a given receiver $R_i \in \Omega$, where $\Omega$ denotes a set of random receivers on an area of size ($x_i \times y_i$), is determined by:

$$D_{E2E}(R_i) = E_T(b, G) \times \frac{E_{Dist}(x_i, y_i)}{r(n)} \tag{14}$$

where $E_{Dist}(x_i, y_i)$ is given by Eq. (12) as:

$$E_{Dist}(x_i, y_i) = \frac{1}{15}\left[\frac{x_i^3}{y_i^2} + \frac{y_i^3}{x_i^2} + \sqrt{x_i^2 + y_i^2}\left(3 - \frac{x_i^2}{y_i^2} - \frac{y_i^2}{x_i^2}\right)\right] + \frac{1}{6}\left[\frac{y_i^2}{x} arcosh\frac{\sqrt{x_i^2 + y_i^2}}{y_i} + \frac{x_i^2}{y_i} arcosh\frac{\sqrt{(x_i^2 + y_i^2)}}{x_i}\right] \tag{15}$$

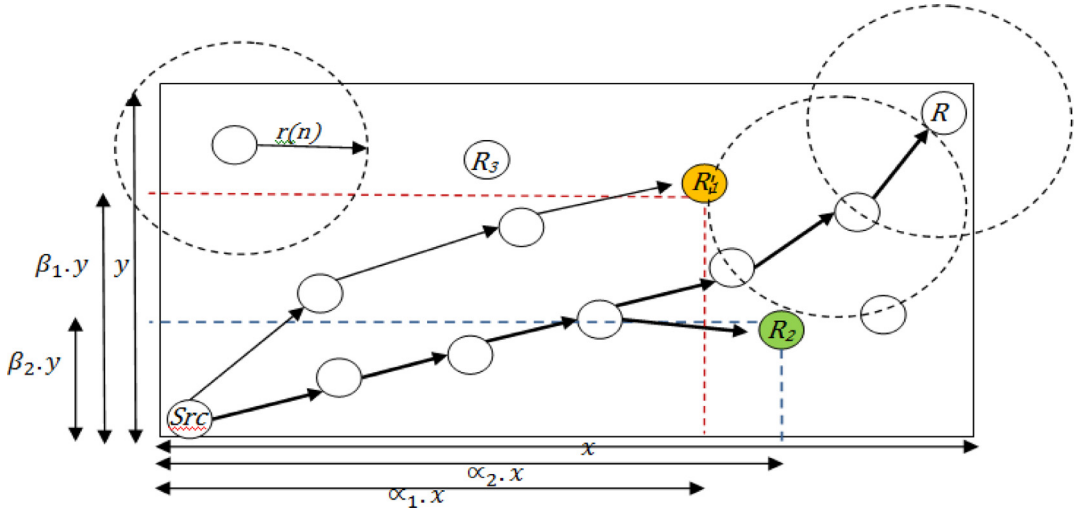



**Fig. 8.** System area of size ($x \times y$).

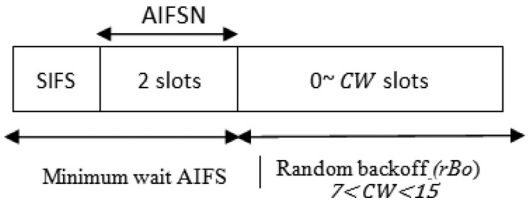



**Fig. 9.** waiting time in the video queue according to the EDCA mechanism.

Thus, from (14) we conclude that the average $D_{E2E}$ for all receivers is written as

$$D_{E2E}(\delta) = \frac{\sum_{i=1}^{\delta} D_{E2E}(R_i)}{\delta} = \frac{\sum_{i=1}^{\delta} (E_T(b,G) \times \frac{E_{Dist}(x_i,y_i)}{r(n)})}{\delta} \quad (16)$$

Let $b = \sum_{j=1}^{G} (b^j)$, be the number of blocks, where $b^j$ is the total number of blocks generated from the $j$th GOP. The average time taken for all blocks, delivered between two successive nodes $n_i$ and $n_{i+1}$ is given by:

$$E_T(b,G) = \frac{1}{G} \times \sum_{j=1}^{G} \frac{\left(\sum_{i=1}^{b^j} e(t)^i\right)}{b^j} \quad (17)$$

where $e(t)^i$ is the time taken by $i$th block expressed by:

$$e(t)^i = t_s^i + \sigma + t_r^i \quad (18)$$

where $t_s^i$, $\sigma$ and $t_r^i$ are the transmission, propagation and receiving delays for the $i$th successful transmitted block for one hope. In particular, we define $t_s$ as the sum of the coding delay and the time period during which the medium is sensed busy:

$$t_s^i = d_c^i + AIFS \quad (19)$$

In fact, when there is a block ready for transmission at the MAC queue, the node senses the channel to be *idle* for a minimum waiting time AIFS and a random backoff time, noted as *rBo* (see Fig. 9). Hence, the AIFS is determined as follow:

$$AIFS = SIFS + AIFSN \times T_{slot} + rBo \times T_{slot} \quad (20)$$

where both EDCA parameters SIFS and AIFSN present the length of the shortest interframe space and the AC-specific AIFS number, respectively. $T_{slot}$ is the duration of time slot. The random backoff must verify the expression given by

$$rBo \in [0, CW]; \quad CW_{min} \le CW \le CW_{max}, \quad (21)$$

where CW is the contention window.

Let $X_i(b_{sz}^j)$ denotes the number of time slot that takes the $i$th block of $b_{sz}^j$ packets to be en-coded. Then the coding delay, for a given $b_{sz}^j$, denoted as $d_c^i(b_{sz}^j)$ is the time required to code the $i$th block by one relay node of the $j$th GOP. Hence we have

$$d_c^i(b_{sz}^j) = X_i(b_{sz}^j) \times T_{slot} \quad (22)$$

Hence, using (20) and (22), (19) can be written as follow:

$$t_s^i = X_i(b_{sz}^j) \times T_{slot} + SIFS + AIFSN + rBo \times T_{slot} \quad (23)$$

Moreover, we define $t_r$ as

$$t_r^i = Y_i(b_{sz}^j) \times T_{slot} \quad (24)$$

where $Y_i(b_{sz}^j)$ is the number of time slots that takes the $i$th block of $b_{sz}^j$ packets to be decoded. Then the decoding delay, for a given $b_{sz}^j$, denoted as $d_d^i(b_{sz}^j)$ is the time required to decode the $i$th block by one relay node of the $j$th GOP. Hence:

$$d_d^i(b_{sz}^j) = Y_i(b_{sz}^j) \times T_{slot} \quad (25)$$

Therefore, the time taken by each block can be evaluated by plugging (23) and (24) into (18) which yields:

$$e(t)^i = T_{slot}\left[X_i(b_{sz}^j) + AIFSN + rBo + Y_i(b_{sz}^j)\right] + SIFS + \sigma \quad (26)$$

Thus, Eq. (17) is presented by:

$$E_T(b,G) = \frac{1}{G}\sum_{j=1}^{G} \frac{\sum_{i=1}^{b^j} e(t)^i}{g^j} b_{sz}^j = \frac{1}{G}\sum_{j=1}^{G}\left(\frac{b_{sz}^j}{g^j}\sum_{i=1}^{b^j}\left[T_{slot}(X_i(b_{sz}^j) + AIFSN + rBo + Y_i(b_{sz}^j)) + SIFS + \sigma\right]\right) \quad (27)$$

Finally, using (27) and (15) the average $D_{E2E}$ for random receivers defined in (16) is expressed by:

$$\begin{aligned} D_{E2E}(\delta) &= \frac{\sum_{i=1}^{\delta} D_{E2E}(R_i)}{\delta} \\ &= \frac{\sum_{i=1}^{\delta} (E_T(b,G) \times \frac{E_{Dist}(x_i,y_i)}{r(n)})}{\delta} \\ &= \frac{1}{\delta} \times \sum_{i=1}^{\delta} \frac{1}{r(n)} (E_T(b,G) \times E_{Dist}(x_i,y_i)) \end{aligned} \quad (28)$$

where:

$$E_T(b,G) = \frac{1}{G} \sum_{j=1}^{G} \frac{\sum_{i=1}^{b^j} e(t)^i}{g^j} b_{sz}^j$$

$$= \frac{1}{G} \sum_{j=1}^{G} \left( \frac{b_{sz}^j}{g^j} \sum_{i=1}^{b^j} \left[ T_{slot} \left( X_i(b_{sz}^j) + AIFSN + rBo + Y_i(b_{sz}^j) \right) + SIFS + \sigma \right] \right)$$

and

$$E_{Dist}(x_i,y_i) = \frac{1}{15} \left[ \frac{x_i^3}{y_i^2} + \frac{y_i^3}{x_i^2} + \sqrt{x_i^2+y_i^2} \left( 3 - \frac{x_i^2}{y_i^2} - \frac{y_i^2}{x_i^2} \right) \right] + \frac{1}{6} \left[ \frac{y_i^2}{x} arcosh \frac{\sqrt{x_i^2+y_i^2}}{y_i} + \frac{x_i^2}{y_i} arcosh \frac{\sqrt{(x_i^2+y_i^2)}}{x_i} \right]$$

### 5.2. Throughput

The throughput is said achievable if *Src* can send at a rate of at least $\tau$ bits per second to a set of destinations. Let $N(R_i, t)$ be the number of packets transferred by the source node and successfully received by receiver $R_i$ in $t$ timeslots. Then the average throughput $\tau$ is [20]:

$$\tau(\delta,t) = \frac{1}{\delta} \sum_{i=1}^{\delta} \lim_{t \to +\infty} \frac{l \times N(R_i,t)}{t} \quad (29)$$

where $l$ is the size of the packet and $N(R_i, t)$ is expressed as:

$$N(R_i,t) = N(R_i,t) \times P(r(n)), \quad (30)$$

where $P(r(n))$ is the probability that one packet is successfully received in time. $P(r(n))$ is defined as in [21]:

$$P(r(n)) = P_{H_1}(r(n)) \cdot P_{H_2}(r(n)) \cdot P_{H_3}(r(n)) \quad (31)$$

where $P_{H1}(r(n))$ is the probability that no hidden terminal is in the transmitting state. Under the condition that $H_1$ is true, we have $P_{H_2}(r(n))$ is the probability that no node within the area of $S(r(n))$ transmitting during the duration $T$. $P_{H_3}(r(n))$ is the probability that no nodes in $r(n)$ starts transmission during the slot that collides with the transmission of the node $n$.

### 5.3. Average packet delivery rate (APDR)

It is important to highlight that PDR measures the ratio of successfully received packets. Hence, higher PDR means better performance. Note that PDR is related to the packet loss rate by the following relation $PDR = 1 - packet\ loss\ rate$. The average packet delivery ratio (APDR) of a multicast destination is defined as the ratio of the number of packets received by all destinations and the total number of packets sent by the source. The packet loss is due to the dropped packets from the queues. In fact, if the queue is full, packets from the same flow may be lost. Moreover, if the waiting time of each packet before being encoded is expired, this packet is dropped from the queue. In addition, if the relay node cannot decode the coded packets, these packets will be dropped immediately from the queue.

$$APDR = \frac{1}{\delta} \sum_{i=1}^{\delta} \frac{N(R_i)}{N}, \quad (32)$$

where $N$ is defined as $N = N_{BL} + N_{EL}$ initially sent by the source node *Src* and $N(R_i)$ is written as:

$$N(R_i) = (N_{BL} + N_{EL}) \times P(r(n)) \quad (33)$$

Substituting (30) and (33) into (32), we obtain:

$$APDR = \frac{1}{\delta} \sum_{i=1}^{\delta} \frac{(N_{BL} + N_{EL}) \times P(r(n))}{N_{BL} + N_{EL}} \quad (34)$$

### 5.4. Average peak signal to noise ratio

The average Y-PSNR for a multicast destination is the PSNR for the luminance component Y in the YUV color space. In general, given an "original" and a "reconstructed" video frame $n$ with $w \times h$ resolution, for any component c the c–PSNR (this work considers the Y component) is computed as:

$$Average\ Y - PSNR = \sum_{n=1}^{n_f} (Y - PSNR_n) \quad (35)$$

where $Y - PSNR_n$ is expressed by:

$$Y - PSNR_n = \frac{1}{\delta} \sum_{l=1}^{\delta} 10 \times \log \left( \frac{255^2}{MSE_n^l} \right), n \in \{1, \ldots, n_f\} \quad (36)$$

where $MSE_n$, the Mean Square Error, for the n*th* frame is defined by:

$$MSE_n = \sum_{i=1}^{w} \sum_{1}^{h} \left( \frac{|A_n^l(i,j) - B_n^l(i,j)|^2}{w \times h} \right), n \in \{1, \ldots, n_f\} \quad (37)$$

where $w$ and $h$ define the width and the height of the frame respectively. Thus, $w \times h$ are the number of pixels. $n_f$ is the number of frames (in our case $n_f$=300); $i$, $j$ are the number of rows and columns in the input images, respectively; $A_n^l(i,j)$ is the value of the Y-component for the pixel $(i, j)$ in the n*th* frame of the original video corresponding to the l*th* receiver; and $B_n^l(i,j)$ is the value of the Y-component for the pixel $(i, j)$ in the n*th* frame of the reconstructed video corresponding to the l*th* receiver.

### 5.5. Jitter

The Jitter is measured as the difference in delays between two consecutive SVC video packets of the same video sequence. In the simulations, the jitter is measured as the average of all jitter values by using the following formula:

$$JITTER = \frac{1}{\delta} \sum_{i=1}^{\delta} \frac{T_c - T_l}{T_d} \quad (38)$$

where $T_c$ and $T_l$ are the received times of the current packet and the last packet respectively and $T_d$ is the difference of the sequence number of two consecutive packets.

## 6. EMSCNC performance evaluation

In order to validate our analytical model, we carry out extensive simulations using ns-2 simulator.

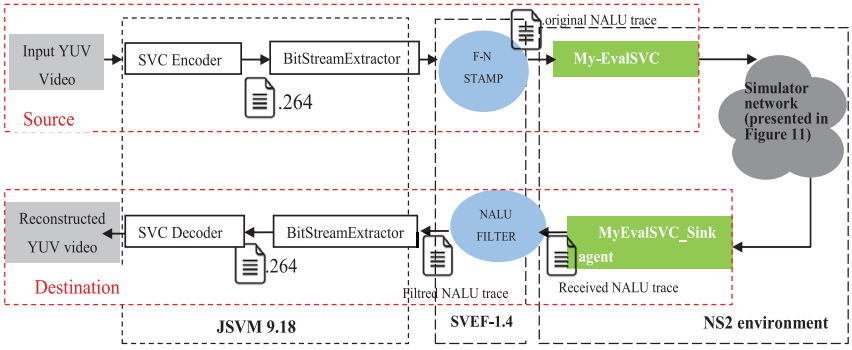



**Fig. 10.** Simulation environment for video transmission with the proposed algorithms over MANET.

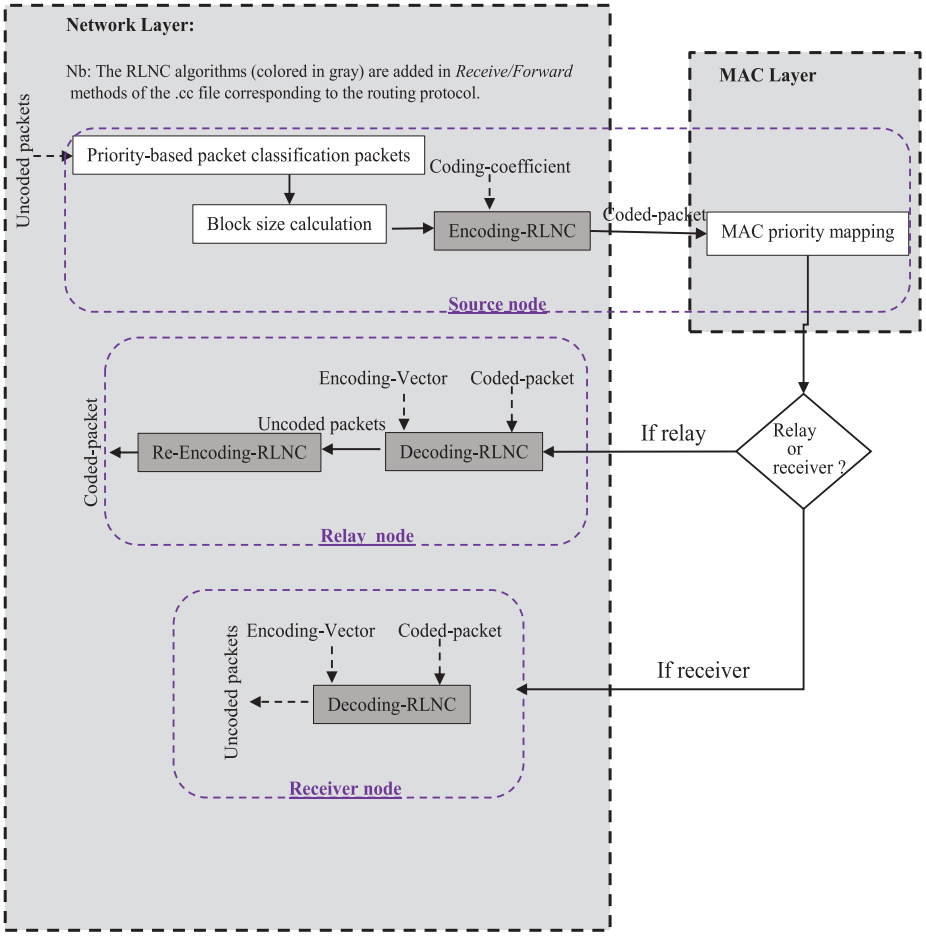



**Fig. 11.** EMCSNC process under NS-2 simulator.

### 6.1. Simulation environment

We consider the network environment shown in Fig. 10. Video streams are coded with version JSVM 9.18 of the SVC reference software.

Fig. 10 details the SVC generator traffic. In fact, the NS-2 simulator works with scalable video coding traffic (SVC) Fig. 11. The JSVM software encodes YUV video with different encoding parameters (temporal scalability, spatial scalability, quality scalability) [2]. This work considers temporal scalability. Next, BitStreamExtractor (provided by JSVM software) and F-N Stamp (provided by SVEF) are used to generate an original NALU Trace file. In the header structure of each SVC packet, three identifiers are added depending on the type of scalability, namely DID, TID, and QID for spatial, temporal and quality scalability respectively. Since only the temporal scalability is considered, TID is modified according to the packet priority level. Packets are encoded according to three levels, as detailed previously. Thus packets belonging to I or P frames have a triplet equal to (0, 0, 0). SVC packets belonging to the first enhancement layer have a triplet set to (0, 1, 0). Finally packets from the third enhancement layer (i.e., B2 frames) have a (DID, TID, QID) triplet set to (0, 2, 0).

In NS-2, we add a new agent called myEvalSVC to read the trace file generated by the SVEF framework. Then, NS-2 generates a new trace file with the corresponding packets in the designated time. At the receiver side, an agent called myEvalSVC_Sink is used to receive the SVC packets and record the related information, such as receiving time, packet size, frame number, and so on. This

**Table 4**
Simulation parameters.

| Parameters | values |
|---|---|
| Video in CIF (352 × 288) | Foreman.yuv |
| Videos in HD (1280 × 720) | Stockholm.yuv |
| Number of frames in video ($N_f$) | 300 |
| Block_size | Dynamic: [3, . . . , 12] |
| Mobility Routing Protocol | Random Waypoint mobility |
| Nodes | 100 |
| packet size | 1500 |
| MAC specification | 802.11e (EDCA) |
| propagation model (physical layer) | two-ray ground model (TRG) |
| Simulation time | 120 s |
| $T_{slot}$ | 20 $\mu$s |
| AIFSN | 2 |
| SIFS | 10 $\mu$s |
| $CW_{min}$ | 7 |
| $CW_{min}$ | 15 |

receiver trace file is then fed into NALU FILTER to generate the Filtered NALU Trace. In this process, late frames or frames that cannot be decoded are eliminated. Following, the Filtered NALU Trace is sent to BitStreamExtractor to generated the Filtered H.264 Video. Finally, this Filtered H.264 Video is decoded by JSVM Decoder to generate the reconstructed YUV Video.

Simulations consider 100 nodes following the Random-Waypoint mobility model with 0 pause time, varying speed, placed in an area of (1500 × 2000)m$^2$. The ns2 simulator implements three propagation models to simulate wireless ad hoc networks which are Free Space model, two ray ground model and shadowing model. In our work, propagation on the physical channel is simulated using the two-ray ground model (TRG) which is an enhanced version of the Free Space model. It considers the direct path and the ground reflection. According to this model sender and receiver are assumed to be in Line Of Sight (LOS). Moreover, it is assumed that all the nodes offer MAC level QoS according to the IEEE 802.11e standard which is applied as a patch in NS-2. At the network layer, we restrict our attention to employ PUMA as the multicast routing protocol. The considered topology has one source node and 10 receivers. Video streaming is divided into different layers that provide different degrees of quality. The source node transmits three temporal SVC-layers coded video streams. This delivered video is a "Foreman" CIF format sequence composed of 300 frames, coded at 30 *frames*/s with a GOP of 8. The bit rate of the video sequences is assumed constant and it is set to 2 *Mbps*. All the measurements are made over a period of 120 s. In order to generate statistically quality measures, each metric is averaged over 10 simulation runs with various random seeds. Simulation parameters are summarized in Table 4.

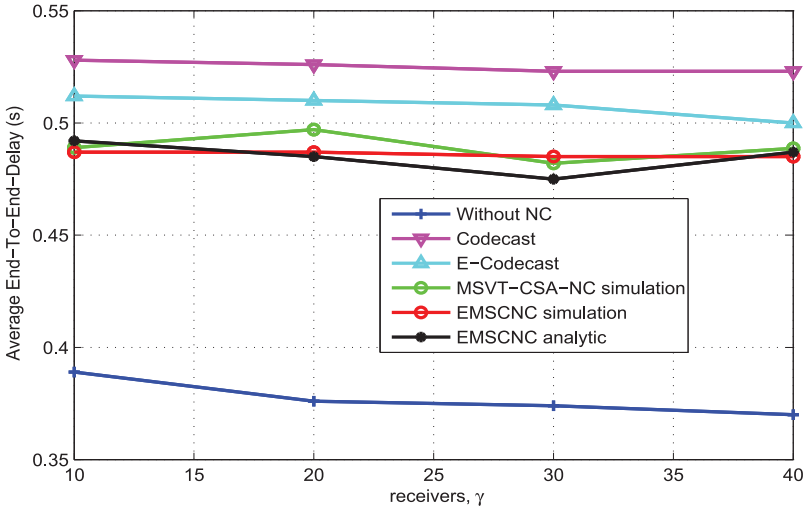



**Fig. 12.** Average delay versus $\delta$ receivers, when *Speed* = 20m/s.

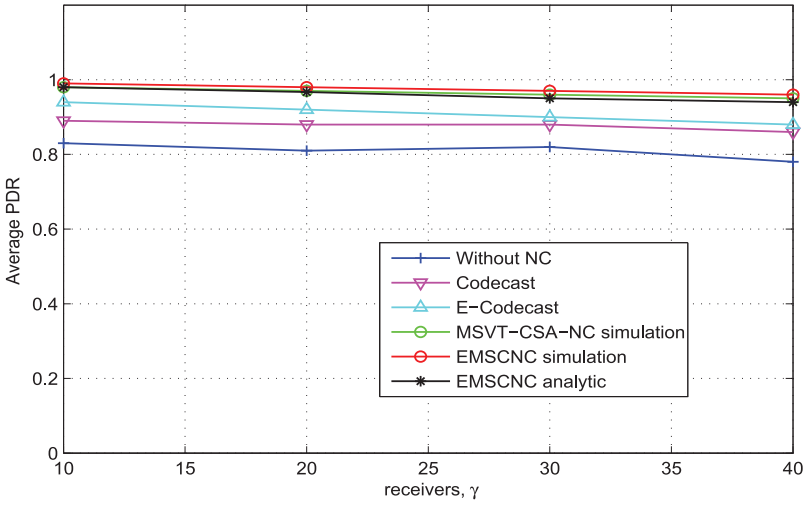



**Fig. 13.** Average PDR versus $\delta$ receivers when *Speed* = 20m/s.

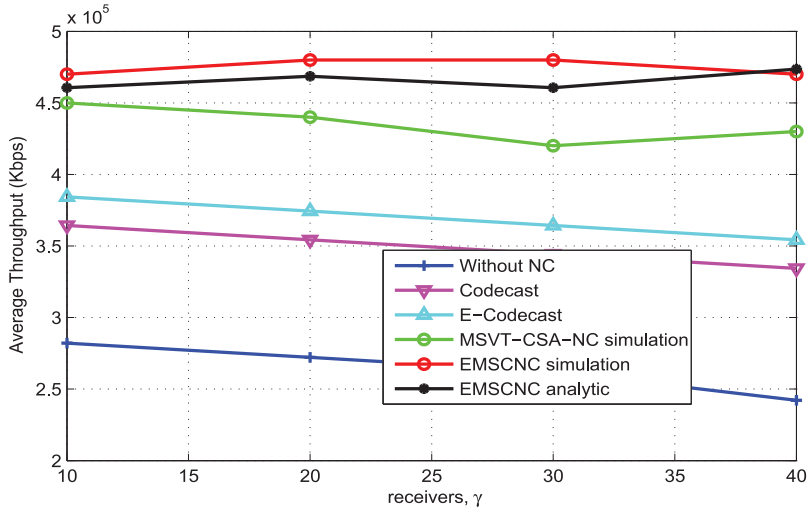



**Fig. 14.** Average throughput versus $\delta$ receivers when $Speed = 20\text{m/s}$.

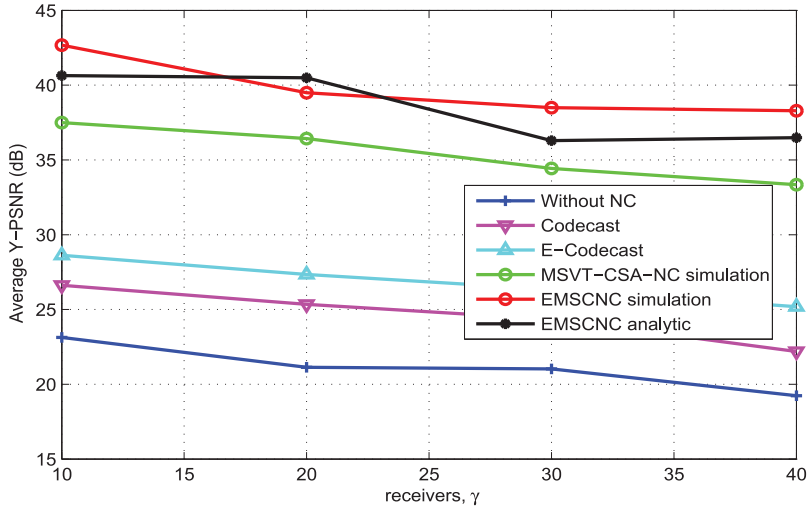



**Fig. 15.** Average Y-PSNR versus number of receivers for CIF format.

### 6.2. Simulation results

According to the proposed approach and the analytical model described in Sections 3 and 4 respectively, Fig. 12 illustrates the average $D_{E2E}$ by comparing the performance of EMSCNC for both analytical and simulation curves. Results demonstrates clearly that the proposed analytical model is accurate and close to simulation results. In fact, while the average delay obtained analytically is 0.487s, the average delay obtained by simulation is 0.475s. Moreover, Fig. 12 compares the delay for EMSCNC, MSVT_CSA_NC [1], Codecast and the conventional multicast transmission without NC, in MANET. Results show that our proposed schemes EMSCNC and MSVT_CSA_NC outperforms the other state-of-the-art NC-based schemes such as Codecast and E-Codecast. In fact, when high priority packets drop increases, more retransmissions are required and hence the $D_{E2E}$ increases. In addition, each node in IEEE 802.11e EDCA mode has 4 queues with different priority levels.The waiting time of a packet in a 4-queues node is less than the waiting time in a 1-queue node. Therefore, the associated queuing delay decreases. Also, as expected, using NC increases the average $D_{E2E}$. This increase is the price of improving the other performance metrics (throughput, PDR, PSNR) as shown in the next figures.

Fig. 13 displays the average PDR for four different cases with varying number of receiver nodes. It is important to remember that higher PDR means better performance since $PDR = 1 - packet\ loss\ rate$. Our proposed scheme outperforms the others. EMSCNC and MSVT_CSA_NC provide better and more consistent results than other approaches. In fact, for 20 receivers, EMSCNC allows an average PDR close to 100%. Whereas, the average PDR of the conventional multicast represented by PUMA degrades to 81%. In addition, we observe a close agreement between the analytical and simulation results. Values obtained by simulation for EMSCNC are slightly higher than the analytical ones.

Fig. 14 plots the average throughput (obtained analytically and by simulation) of our proposed scheme (EMSCNC) and other schemes for different numbers of receiver nodes. By simulations, we find that EMSCNC offers the largest average throughput. We notice that using NC improves greatly the average throughput. Also, analytical results for EMSCNC match well with the simulation ones.

In Fig. 15, showing the PSNR versus number of receivers of "Foreman" sequence, we see again that EMSCNC outperforms other schemes. We can see that the average PSNR values for simulation results are close to the analytical PSNR values. It should be noted that our scheme is more efficient than the other schemes. In fact,

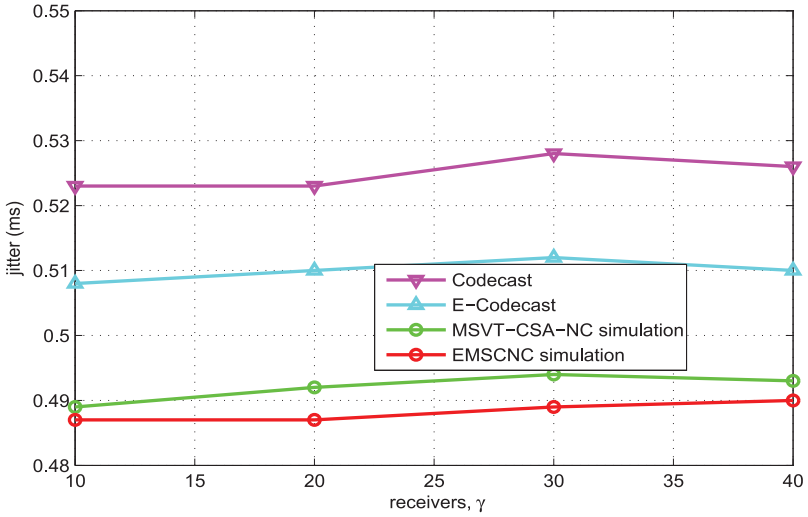



**Fig. 16.** Average jitter for different NC-based schemes.

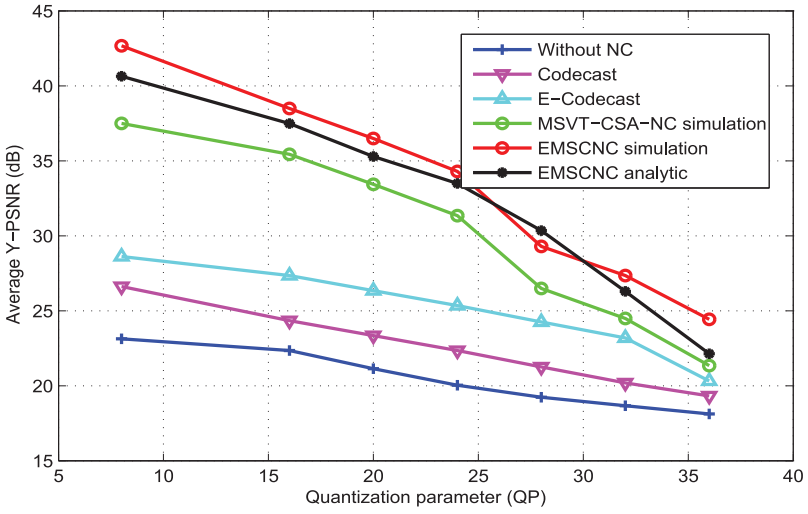



**Fig. 17.** Average Y-PSNR as a function of QP when $\delta = 10$ for CIF format.

from Fig. 15 the average PSNR gives a value of 42.64*dB* for EMSCNC; while Codecast and E-Codecast [14], for instance, gives a value of 26.64*dB* and 27.85*dB*, respectively.

In order to better validate the effectiveness of the proposed scheme compared to other NC-based schemes, we evaluate the Jitter as a key metric to characterize the quality of service (QoS) in wireless networks. Fig. 16 shows the jitter values for NC-based schemes (EMSCNC, Codecast, E-Codecast, MSVT_CSA_NC). It demonstrates that EMSCNC outperforms other state-of-the-art NC-based schemes in terms of jitter. In fact, EMSCNC reduces the average jitter of the decoded video streams from 0.523 for Codecast to 0.485.

Furthermore, to convince the analytical model and performance evaluation, we put the analytical and simulation results after encoding the sequence video at different quantization parameters values (QP). Simulation results for various QP values are presented in Fig. 17. These graphs show that lower quantization parameter achieves higher video quality getting from higher PSNR values. Thus, as QP value increases, it is observed that average PSNR decreases. In fact, when QP = 16 the reconstructed pictures for EMSCNC have an acceptable video quality with PSNR values greater than 37*dB*. Whereas when QP is set to 32 the average PSNR decrease considerably until 27*dB*.

As HD is becoming popular, we consider the HD resolution videos in our experiments. In fact, we consider Stockholm sequence as the HD format sequence used testing. Fig. 18 displays the average PSNR for a fixed QP = 8. Results show that our scheme outperforms the other schemes. Whereas compared to CIF format, the average PSNR is decreased, this is due to the fact that H.264/SVC codec is not adequate for HD format video. Authors in [22] demonstrates that the codec High-Efficiency Video Coding (HEVC) is especially efficient for coding high-resolution video such as high-definition (HD) video.

Video quality and stringent delay requirements pose challenges in real-time video transmission. Our scheme improves the performance of multimedia streaming. We have shown, through simulations and analytical results, that our scheme is able to adapt well video streaming and to provide stable performance in terms of E2ED, PDR, PSNR, Jitter and throughput. Particularly, simulation and analytical results show that our approach is attractive in comparison to Codecast (see Table 5).

We highlighted that all NC-based schemes introduce performance improvement at the price of delay increase. Also, our scheme introduces a slight increase of the average $D_{E2E}$ compared to non-NC based schemes, which presents its only counterpart. Even though, the $D_{E2E}$ provided by our scheme is close to the non-

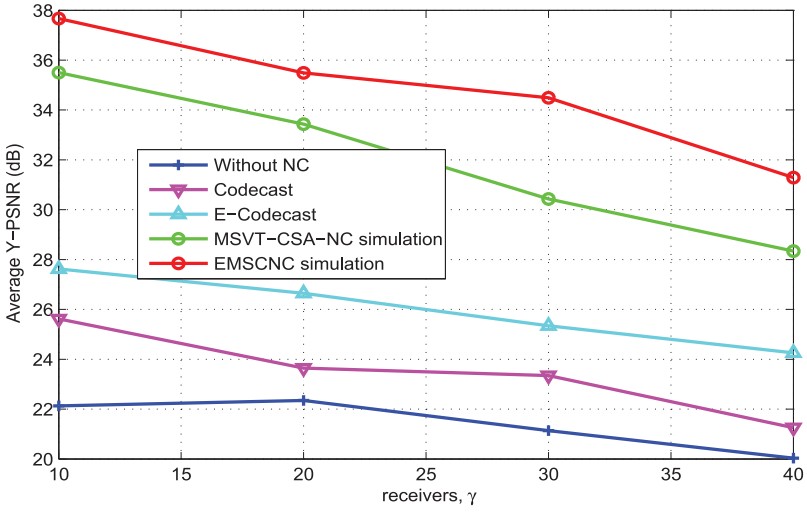



**Fig. 18.** Average Y-PSNR versus $\delta$ receivers for HD format.

**Table 5**
Advantages and the limitation of our scheme compared to other approaches. Legend: – negative effect , + Small effect, ++ Medium effect, +++ Large effect.

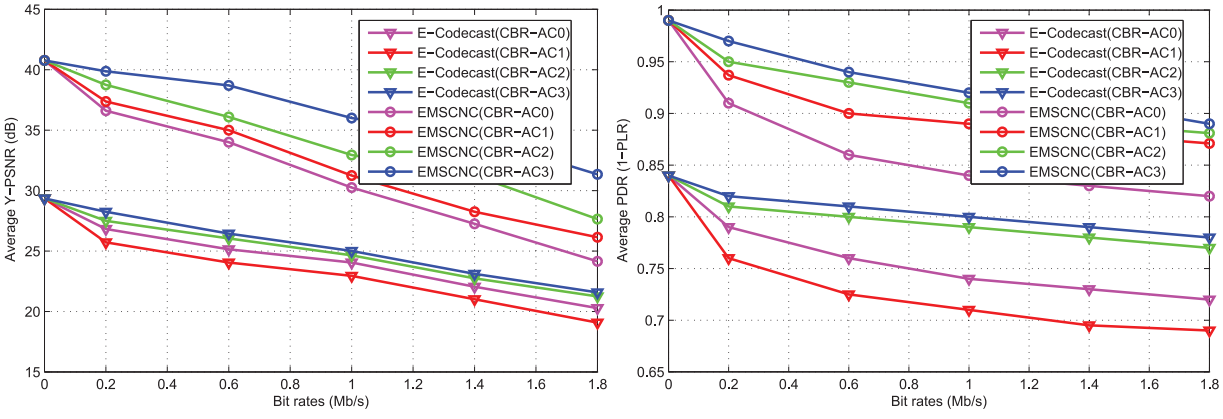
| | E2ED | PDR | PSNR | Throughput |
|---|---|---|---|---|
| Without NC | +++ | – | – | + |
| Codecast | – | ++ | ++ | ++ |
| MSVT_CSA_NC | + | ++ | ++ | ++ |
| EMSCNC | ++ | +++ | +++ | +++ |

NC based schemes and still acceptable. We have also shown that our scheme is able to provide stable performance in terms of $D_{E2E}$ compared to Codecast and MSVT_CSA_NC. Our proposed EMSCNC has the lowest end to end delay between NC-based schemes (as shown in Fig. 12 for instance) thanks to higher priority mapping of ACs. This is because the highest priority AC has the shortest Arbitration Inter Frame Space (AIFS) allowing encoded packets to quickly access the channel.

*6.2.1. EMSCNC performance in presence of background traffic*

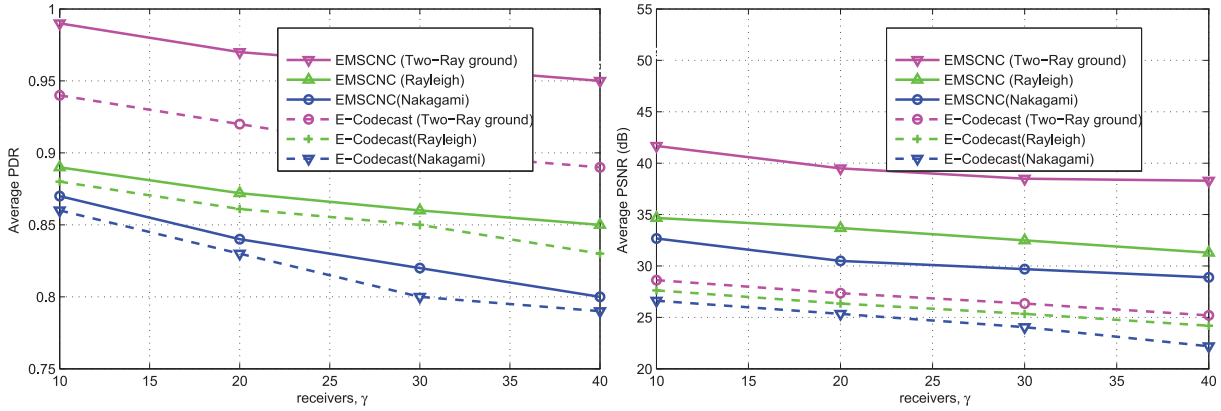
To have a better view on the performance of the proposed EMSCNC scheme, we evaluate it in presence of background traffic (i.e, CBR traffic) running over IEEE 802.11e networks in different priority classes. In particular, we consider a scenario where a wireless source multimedia node is transmitting encoded H.264/SVC and CBR traffic to ten receivers, all happening simultaneously. Thus, this simulation studies the impact of background traffic when it is mapped to different access categories. In fact, we consider four CBR mapping cases, namely when CBR traffic is mapped to: i) AC3, ii) AC2, iii) AC1, iiii) and the worst case when CBR is mapped to AC0 (having the highest priority). The Packet Delivery Ratio (PDR) and PSNR are used as the basic video quality evaluation. PDR measures the ratio of the number of successfully received packets to the total number of sent packets. Hence, higher PDR means better performance. It is also important to note that PDR is related to the packet loss rate by the following relation PDR = **1** – packet_loss_rate.

Figs. 19a and 19b plot respectively PSNR and PDR values for Foreman encoded video sequence and transmitted from the same multimedia source. We also propose to evaluate E-Codecast scheme in presence of background traffic and compare it with EMSCNC scheme to show that our proposed scheme always provide better PDR and PSNR due to its efficient NC design. Several conclusions may be drawn from these figures. Results show that the background traffic generated at the source node increases the virtual collisions that occur at the MAC layer. We remark that background traffic lead to decreases the perceptual quality, depicted by lower PSNR values and lower PDR. We also




(a) PSNR in presence of background traffic (b) PDR in presence of background traffic

**Fig. 19.** Performance of EMSCNC scheme in presence of background traffic (CBR traffic is assigned to AC0, AC1, AC2 or AC3).




(a) PDR for EMSCN scheme using different propagation models

(b) PSNR for EMSCN scheme using different propagation models

**Fig. 20.** Performance of EMSCNC and E-Codecast schemes using different propagation models (the two-Ray ground, Rayleigh, and Nakagami).

notice that our proposed scheme always outperforms E-Codecast, thanks to the fact that our scheme forms blocks based on packet priorities using an inter-layer compensation algorithm. In addition, the choice of block size in EMSCNC scheme is based on the number of high priority packets which is dynamically updated after each decoding step.

The video quality decreases when the background traffic bit-rate increases because E-Codecast gives equal priority to all packets with no packet differentiation at the MAC layer. Accordingly, there is high probability of dropping the Intra coded packets (from I-Frames). Indeed, when mapping the background traffic to AC1, the overall image quality decreases due to the lower efficiency of E-Codecast from 28.52dB at 0.2Mb/s to 19.08dB at 1.8Mb/s.

*6.2.2. EMSCNC performance under different propagation models*

To study the impact of propagation model on EMSCNC performance, we consider three radio propagation models for MANET communication using NS2 simulator, namely: two-Ray ground, Rayleigh, and Nakagami [26]. PDR and PSNR are evaluated and shown in Fig. 20. We notice, from Fig. 20a, that the two-Ray ground model allows to deliver more packets, followed by Rayleigh, and then Nakagami model. It is to be reminded that PDR measures the ratio of successfully received SVC packets. Similarly, we notice from Fig. 20b that our proposed scheme provides higher PSNR considering the two-ray ground than the other propagation models (Rayleigh and Nakagami). Thus, simulation results indicate that the propagation model has similar impact on the performance of NC-based schemes (e.g, E-Codecast). Even though, the propagation model has an impact on the used transmission scheme, the performance improvement coming from the utilization of our proposed scheme, compared to other NC-based schemes, does not depend on the propagation model. Previous simulations consider the simple TwoRayGround model since it is commonly used for MANET as in [23–25].

## 7. Conclusion

In this paper, we proposed a novel cross layer approach named EMSCNC for modeling network coding in a wireless network. EMSCNC requires two workflows associated respectively to the source-node and the Intermediate-node. It aims to improve the performance of multicast transmission for video streaming in MANET. It can significantly improve the PDR, PSNR, end-to-end-delay, Jitter, throughput and hence the QoS of multicast video. Additionally, the proposed scheme provides significant improvement on network performance under different receivers. We also examined the effect of QP variation on the delay and received video quality. Moreover, as HD is becoming popular, we consider the HD resolution videos in our experiments. Results show that EMSCNC outperforms other NC schemes.

Furthermore, we have proposed an analytical model to evaluate the performance for the average delay, throughput, PSNR and PDR for EMSCNC. Results show a close agreement between the performance evaluated by the analytical model and by simulations of EMSCNC.

As an extension of this work, we will investigate the utilization of EMSCNC in multicast wireless network with multi source nodes using a variety of sequence videos. Moreover, H.264/SVC is open source. Hence, we will implement our new framework in some mobile devices and evaluate the performance of the proposed approach in real environment. In addition, we will integrate the HEVC encoded video to investigate the pertinence of the proposed EMSCNC.

## References

[1] O. Ben Rhaiem, L. Chaari Fourati, W. Ajib, Qos improvement for video streaming over MANET using network-coding, in: Proceedings of Vehicular Technology Conference (IEEE VTC Fall), 6-9 Sept, 2015.
[2] O. Ben Rhaiem, L. Chaari Fourati, H.264/SVC scalability performance analysis, in: International Conference on Signal and Image Processing Applications (ICSIPA), 2013, pp. 203–208. 8-10 Oct.
[3] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable extension of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 17 (9) (2007) 1103–1120.
[4] J.F. Kurose, K.W. Ross, Computer Networking-A Top-Down Approach Featuring the Internet, Pearson Education Inc, James Francis Kurose, 2005.
[5] S. Zhang, S.C. Liew, P.P. Lam, Hot topic: physical-layer network coding, Proceedings of the 12th annual International Conference on Mobile Computing and Netw., ser. MobiCom '06, ACM, 2006, pp. 358–365.
[6] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, J. Crowcroft, Xors in the air: practical wireless network coding, IEEE/ACM Trans. Netw. 16 (3) (June 2008) 497–510.
[7] R. Ahlswede, N. Cai, S.Y. Li, R. Yeung, Network information flow, IEEE Trans. Inf. Theory 46 (4) (Jul 2000) 1204–1216.
[8] S.y.R. Li, S. Member, R.W. Yeung, N. Cai, Linear network coding, IEEE Trans. Inf. Theory 49 (2003) 371–381.
[9] R. Koetter, M. Médard, S. Member, An algebraic approach TC network coding, IEEE/ACM Trans. Netw. 11 (2003) 782–795.
[10] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, IEEE Trans. Inf. Theory 52 (10) (Oct 2006) 4413–4430.
[11] P.A. Chou, Y. Wu, K. Jain, Practical network coding, in: Proceedings of Allerton Conference on Communication, Control, and Computing, October, 2003. Invited paper.
[12] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the h.264/avc standard, IEEE Trans. Circuits Syst. Video Technol. 17 (9) (Sept 2007) 1103–1120.
[13] J.-S. Park, M. Gerla, D. Lun, Y. Yi, M. Medard, Codecast: a network-coding-based ad hoc multicast protocol, IEEE Wireless Commun. 13 (5) (October 2007) 76–81.

[14] Yang, J. Bachmatiuk, V.N. Talooki, J. Rodriguez, Improved codecast for SVC video multicast, in: Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2013 IEEE 18th International Workshop on 25-27 Sept., 2013, pp. 13–17.
[15] G. Tan, X. Ni, X. Liu, C. Qu, L. Tang, Real-time multicast with network coding in mobile ad-hoc networks, Intell. Autom. Soft Comput. 18 (7) (2012) 783–794.
[16] R. Vaishampayan, J. Garcia-Luna-Aceves, Efficient and robust multicast routing in mobile ad hoc networks, in: Proceedings IEEE International Conference Mobile Ad-hoc and Sensor Systems, Oct 2004, pp. 304–313.
[17] G. Tan, X. Peng, X. Ni, B.A.F. Lin, X. Liu, Pncrm: a novel real-time multicast scheme in manets based on partial network coding, J. Netw. 8 (10) (2013).
[18] D. Wang, Q. Zhang, J. Liu, Partial network coding: theory and application for continuous sensor data collection, in: Proceedings IEEE International Workshop Quality of Service, June, 2006, pp. 93–101.
[19] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Commun. Mobile Comput. J. 2 (5) (2002) 483–502.
[20] P.S. David, A. Kumar, Network coding for TCP throughput enhancement over a multi-hop wireless network, in: Proceedings of International Conference on Communication Systems Software and MiddlewareCOMSWARE, 6-10 Jan, 2008.
[21] X. Ma, Packet reception ratios in two-dimensional broadcast ad hoc networks, in: Proceedings of International Conference on Computing, Networking and Commun. (ICNC2012), Jan. 30 -Feb. 2, 2012.
[22] X. Lian, W. Zhou, Z. Duan, R. Li, An efficient interpolation filter VLSI architecture for HEVC standard, in: Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on 9-13 July, 2014, p. 384-388.
[23] P.K. Singh, K. Lego, Comparative study of radio propagation and mobility models in vehicular adhoc NETwork, Int. J. Comput. Appl. 16 (8) (Feb. 2011) 37–42.
[24] A.M. Kanthe, D. Simunic, R. Prasad, Effects of propagation models on AODV in mobile ad-hoc networks, Wireless Pers. Commun. 79 (1) (2014) 389–403.
[25] I.K. Eltahir, The impact of different radio propagation models for mobile ad hoc NETworks (MANET) in urban area environment, in: Wireless Broadband and Ultra Wideband Communications, 2007. Aus. Wireless 2007. The 2nd International Conference on, Sydney, NSW, 2007. 30-30.
[26] F.J. Martinez, C.-K. Toh, J.-C. Cano, C.T. Calafate, P. Manzoni, Realistic radio propagation models (RPMs) for VANET simulations, in: IEEE, Wireless Communications and Networking Conference, April, 2009.

**OLfa Ben Rhaiem** was born in SFAX, TUNISIA, in 1987. She received her master degree in computer science from multimedia and informatics higher institute, SFAX University, TUNISIA (ISIMS), Tunisia, in 2012. Since January 2013, she has been a Ph.D. student in the National Engineering School of Sfax. She is a researcher at Electronics and Information Technology Laboratory (LETI). Her current research interests are content centric network (CCN), network coding (NC), and robust decoding (RD) related to video streaming transmission over wireless networks.

**Lamia Chaari Fourati** was born in Sfax, Tunisia, in 1972. She received the engineering and Ph.D. degrees in electrical and electronic engineering from SFAX National Engineering School (ENIS) in TUNISIA. She received in 2011 HDR telecommunications from ENIS. Actually she is an associate professor at multimedia and informatics higher institute, SFAX University, TUNISIA. She is also a researcher at electronic and technology information laboratory (LETI). Her scope of research are new generation networks, communications, wireless networking, multiple access and MAC design, QoS provisionning, IoT, M2M, SDN, ICN. She is the author or co-author of many journal papers and conferences papers in these areas.

**Wessam Ajib** received an engineering diploma from Institut National Polytechnique de Grenoble, France (INPG-ENSPG) in physical instruments, in 1996, a DEA (Diplôme d'Études Approfondies) from École Nationale Supérieure des Télécommunication (ENST), Paris, France in 1997 in digital communication systems, and a Ph.D. degree in computer sciences and computer networks from ENST, Paris, France, in 2000. He had been an architect and radio network designer at Nortel Networks, Ottawa, ON, Canada between October 2000 and June 2004. He had conducted many projects and introduced different innovative solutions for UMTS system, the third generation of wireless cellular networks. He followed a post-doc fellowship at Electrical Engineering department of École Polytechnique de Montréal, QC, Canada between June 2004 and June 2005. Since June 2005, He had been with the Department of Computer Sciences, Université du Québec à Montréal, QC, Canada, where he is presently a full professor of computer networks. His research interests include wireless communications and wireless networks, multiple access and MAC design, traffic scheduling 5G and cellular networks and cognitive radio networks. He is the author or co-author of many journal papers and conferences papers in these areas.