EMCOS: Energy-efficient Mechanism for Multimedia Streaming over Cognitive Radio Sensor Networks

Abbas Bradai*, Kamal Singh†, Abderrezak Rachedi‡ and Toufik Ahmed**

* CNRS Grenoble Informatics Laboratory UMR 5217, France
bradai@imag.fr
† Université de Lyon, F-42023, Saint-Etienne, France
kamal.singh@univ-st-etienne.fr
‡ CNRS-LIGM University Paris-Est (UPEM)
rachedi@u-pem.fr
** CNRS-LaBRI University of Bordeaux
tad@labri.fr

Abstract— One of the major challenges for multimedia transmission over multimedia WSN (MWSN) in urban environment is the scarcity of spectrum combined with high radio interference. Such environment makes it difficult to ensure high bandwidth, low delay and low packet losses required for real time multimedia streaming applications. We target a scenario of video surveillance in urban environment which not only requires efficiency of spectrum utilization, but also requires energy efficient mechanisms for the battery operated MWSN nodes. In this paper, we propose a new solution for multimedia transmission over WSNs which uses cognitive radio technology for spectrum efficiency and clustering mechanism for energy efficiency. A video streaming solution is proposed that is called "EMCOS: Energy-efficient Mechanism for Multimedia Streaming over Cognitive Radio Sensor Networks ". EMCOS ensures high quality real time multimedia transmission from one or more sources to a given sink, under different spectrum availability conditions, while efficiently using the energy of the MWSN nodes. First, EMCOS clusters the MWSN nodes into different clusters in order to ensure low energy consumption. Additionally for clustering, EMCOS not only takes into consideration the geographic positions, but it also takes into account the actual and the forecast of the channel availability in order to ensure stable clusters. Once the clusters are built, a cluster head is elected for each cluster in a way which preserves the cluster energy by considering the energy utilization of all cluster members. Further, to ensure the content delivery from the source to the sink, a routing / channel selection mechanism is proposed. The channel selection is based on PU activity forecasts to prevent frequent channel switching. Simulations show that our proposal EMCOS outperforms the two existing pioneering mechanisms called SEARCH and SCEEM. EMCOS outperforms them in terms of providing higher video quality, lower end-to-end transmission delay and lower frame loss ratio under varied spectrum conditions.

Keywords-WSN, cognitive radio, multimedia, Qos, energy

1 INTRODUCTION

Radio communications need efficient and adaptive mechanisms to tackle issues related to inherently challenging radio environment and scarcity of radio spectrum. In this context, cognitive Radio (CR) is becoming a promising technology that enhances the performance of radio communication and improves the efficiency of radio spectrum utilization by enabling flexible and adaptive radio capabilities. Such CR capabilities are enabled by combining artificial intelligence, software-defined radios and opportunistic radio spectrum access schemes. Cognitive Radio was defined by Joseph Mitola [1] as intelligent radio that can autonomously make decisions using gathered information about the radio frequency (RF) environment through model-based reasoning and can also learn and plan according to its past experience. Later, Simon Haykin [1] defined CR as an intelligent wireless communication system capable of being aware of its environment, learning, and adaptively changing its operating parameters in real time for providing reliable communication and efficient utilization of the radio spectrum. Such intelligent radios use model-based reasoning, and they learn and adapt based on their past experiences. This definition was extended to CR as an intelligent wireless communication system whose goal is to provide reliable communication and efficient utilization of the radio spectrum. This is achieved through the CR capabilities of environment awareness, real time learning and real time adaptation of its operating parameters such as transmission power, carrier frequency and modulation, etc.

CR uses opportunistic radio spectrum access techniques such that the unlicensed users, or secondary users (SUs), continuously monitor the presence and activity of the licensed users, or primary users (PUs), to find the spectrum holes. This monitoring is done by different spectrum sensing techniques and spectrum holes are the bands which are not being utilized by PUs at a given time or a given location. The SUs can opportunistically utilize these spectrum holes and they should ensure no interference with the PUs.

Cognitive Radio is a promising communication technology for Wireless Sensor Networks (WSNs) [1]. Its opportunistic communication paradigm suits wireless sensor networks that carry bursty traffic due to their event driven nature. Cognitive Radio can help overcome the problems of collision and excessive contention in WSNs that arise due to the deployment of several sensors connected through radio links. In comparison, conventional WSNs use fixed spectrum allocation policy and their performance is limited due to low processing and communication power of sensor nodes which are typically resource-constrained. However, WSNs operate over unlicensed bands which are becoming saturated due to a significant growth in the number of wireless applications using the same bands. Thus, the challenge is to efficiently utilize the spectrum and this challenge can be addressed by using cognitive radio technology for WSNs.

Using CR technology for WSNs leads to a new paradigm called cognitive radio sensor networks (CRSN). A CRSN consists of wireless cognitive radio sensor nodes, which sense an event and collaboratively communicate the information using radio spectrum in an opportunistic manner. Using CR technology in resource-constrained WSNs improves the spectrum utilization, enables multi-channel capability and overcomes the problems related to dense deployment of WSNs or higher communication performance required in some WSN applications.

One potential application of CRSN is Multimedia Wireless Sensor Networks (MWSN). MWSN consists of sensor nodes having low cost cameras and microphones. Multimedia sensor nodes are used to store, process and transmit video, audio and image data for the applications such as tracking and monitoring.

We consider video surveillance in urban environment as the candidate scenario for multimedia streaming with cognitive radio sensor networks. In this scenario, multimedia sensor nodes equipped with cameras are connected with wireless links in ad hoc manner. One or more of these nodes are active at the same time. They continuously monitor the environment and transmit the captured videos to the sink node. These nodes operate on batteries and hence energy efficiency is an important issue for this system. Moreover, for wireless transmission of video the spectrum should be utilized efficiently as nowadays spectrum in cities is a scarce resource. The licensed spectrum bands (3G, 4G, etc.) are highly utilized and the unlicensed spectrum bands such as ISM bands are becoming highly saturated. That is why we propose an energy efficient approach based on cognitive radio for an efficient utilization of the spectrum.

The key issues and challenges of MWSN are high bandwidth demand, high energy consumption, quality of service (QoS) provisioning, data processing, and compressing techniques, and cross-layer design:

- MWSN requires high bandwidth in order to deliver multimedia content such as a video stream, audio stream or images. Providing high bandwidth can be challenging with conventional WSNs which use fixed spectrum allocation and operate over saturated unlicensed bands.
- Multimedia transmission requires certain QoS guarantees. However, QoS provisioning is very challenging in WSNs as radio links can have variable capacity and delay. Nevertheless, in MWSNs, a certain level of QoS must be achieved for reliable delivery of multimedia content.
- Energy efficiency is very important for MWSN as sensor nodes typically have low energy resources. Thus, mechanisms need to be designed that focus on energy efficient communication to exploit transmission power and spectrum characteristics vs. performance and reliability trade-off.

- Efficient spectrum utilization, as discussed above, is important because of scarce nature of radio spectrum. Thus, a dynamic spectrum allocation scheme is needed for MWSNs that can take into account the application requirements in terms of bandwidth, QoS and traffic load.

CR technology can answer the above requirements of MWSNs. CR can provide extra bandwidth and improve the quality of service. However, such cognitive radio approaches, specific to MWSNs, need to be designed which focus on energy efficient communication to exploit transmission power and spectrum characteristics vs. performance and reliability trade-off. In addition, low cost algorithms need to be designed for spectrum sensing and dynamic spectrum usage. Most of the works in the literature focusing on CRSNs are related to only spectrum sensing [2]-[4]. A few works focus on multimedia transmission over cognitive radio networks [5]-[8], but they do not consider the WSN environment and the related constraints.

In this paper, we apply CR technology for MWSNs and the key contributions of this paper are:

1) New spectrum aware clustering mechanism for CRWSN: The clustering mechanism is spectrum aware and takes into consideration the actual as well as the future expected channel states.
2) New energy aware cluster head election mechanism for CRWSN: The proposed mechanism takes into consideration the residual energy of the candidate cluster head as well as the consumed energy of the elected node.
3) Routing and channel selection mechanism for efficient multimedia delivery over CRWSN: The proposed routing is a classical approach which uses TDMA based intra cluster communication and CSMA based inter cluster communication. While the channel selection mechanism is based on the forecast of the PU activity on different channels

This paper is organized as follows: Section I provides an introduction, the related work is discussed in Section II, Section III proposes our approach. Section IV discusses the performance evaluation of our approach and the paper in concluded in section V.

2 RELATED WORK

Some existing works related to CR [1][2], focus on spectrum sensing, dynamic spectrum access, MAC protocols [3], routing protocols and QoS [4]. Above is just a sampling of the literature and many other works exists for CR for ad hoc networks and cellular networks because it is well studied in literature. However, the research on applying CR to WSNs is still in its early stage. The research solutions proposed for general-purpose CR networks cannot be directly applied to CRSNs because of the unique features of WSNs, such as the limited resources and energy constraints. Thus, newer CR solutions are needed that should be customized for WSNs to account for resource and energy constraints. If we specifically focus on MWSNs then there are many energy efficient approaches, as classified in a survey on MWSN [5][6], but cognitive radio is not considered is those approaches. In particular, using CR with MWSN involves a trade-off between spectrum availability, QoS, energy consumption and resource efficiency.

Some works related to multimedia streaming in CRSNs are described in the following text. The following works address issues related to different layers such as transport, network, MAC and PHY layers. Bicen et al. [7] discuss the principal design challenges for multimedia and delay-sensitive data transport in CRSN. They explore different transport layer protocols from the point of view of multimedia transport in CRSNs. However, other issues like clustering and routing are not tackled. In order to address the problem of high bandwidth demands and QoS requirements, Rehmani et al. [8] proposed channel bonding for CRSNs. The idea is to bond the contiguous channels as compared to channel aggregation which can aggregate non-contiguous channels, but is more complex for WSNs. They proposed the channel bonding idea and pointed out many issues and challenges. However, the algorithms and protocols to perform optimal channel bonding were not studied and were left for future work.

Liang et al. [9] studied the QoS performance in CRSNs. They propose a priority based mechanism in MAC layer. The real time traffic is scheduled in guaranteed time slots and the best effort traffic is served in contention access period. They study the performance using analytical models and show that satisfactory performance can be achieved for real time traffic. An extension of this study is presented in [10]. However, they do not consider the energy consumption and only the case related to one cluster of sensor nodes is analyzed. Multi-clusters, clustering mechanisms and inter cluster communications are not considered.

Clustering is considered by Fard et al. [11], who propose a QoS MAC protocol for wireless multimedia sensor networks using multi-channel approach. The wireless traffic is classified into different classes and different channels are adaptively assigned to different traffic. This approach improves the throughput and delay performance while being energy efficient. However, PU activity is not taken into consideration and each cluster-head is assumed to have multiple radio interfaces, which is not realistic. Wen et al. [12] propose a QoS routing protocol for ad hoc CRNs by considering parameters such as available time, frequency bands, transmission range, error rate, primary user (PU) interruption rate and transmission range. However, they only focus on end-to-end delay.

Zhou et al. [13] propose a distributed scheduling algorithm for video streaming over multi-channel multi-radio and multi-hop wireless networks. This approach develops distributed scheduling schemes with the objective of minimizing the video distortion, considering fairness. The scheduling is formulated as a convex optimization problem, and a solution is proposed by jointly considering channel assignment, rate allocation, and routing. The work considers the trade-off between minimizing video distortion of each stream vs. improving global performance by minimizing network congestion. However, they do not take cognitive radio into consideration. They assume all channels are available and dynamic spectrum access considering PU activity is not taken into account. Moreover, note that energy management is not considered in the above works.

As compared to the above works, we propose a multimedia streaming mechanism for MWSN that considers dynamic spectrum access as well as energy management by dividing nodes into clusters and electing cluster heads. We consider actual and the forecast of the PUs activity and we also target realistic conditions by assuming a single radio interface.

3 NETWORK AND ENERGY MODEL

We assume a network of N SU nodes each having a transmission range *r* and initial energy *e*. Each node is equipped with a single transceiver which has the capability to switch among C transmission channels and a common control channel (CC). The transmission channels are used for content transmission, whereas, the CC channel is used for signaling and coordination between the network nodes. By default, the transceiver is tuned to CC and switches to a transmission channel only in the data transmission stage. Moreover, the transmission channels are shared with *k* PUs which access the channel preemptively.

SUs use energy detection method for spectrum sensing and a non-fading environment is assumed. Thus, the signal received by a SU can be represented as follows [15]:

$$S^{su} = \begin{cases} n(t), & if\ H_0 \\ n(t) + S^{pu}(t), & if\ H_1 \end{cases} \tag{1}$$

Where the null hypothesis H_0 represents the PU idle state, the non-null hypothesis H_1 represents the PU transmission state, $n(t)$ is a zero-mean additive white Gaussian noise (AWGN) and $S^{pu}(t)$ is the PU transmission signal. Therefore, the probability that the SU detects the PU activity (P_d) and the probability of a false alarm (P_f) are computed as [15]:

$$\begin{aligned} P_d &= Pr\{X > \lambda_{pu} | H_1\} \\ P_f &= Pr\{X > \lambda_{pu} | H_0\} \end{aligned} \tag{2}$$

Where X is the decision variable of the energy detection mechanism and λ_{pu} represents the decision energy threshold. A low value of P_d equates to a high probability to not detect the activity of the PU, and consequently a high probability of collision with PU. On the other hand, a high P_f implies low channel utilization because it means that, with a high probability, the given SU falsely estimates PU activity on the channel even when the channel is free (H_0) from it. In our model, $\lambda_{pu} = 0.1$.

In our work, we assume a realistic energy dissipation model for radio communication [15]. The radio unit is composed of *Transmit Electronic*, *Transmission Amplifier* and *Receive Electronics* (see Figure 1). We assume that the Transmit or the Receive circuit consumes E_{elect}=50 nJ/bit, and the amplifier consumes E_{ampl}=100 pJ/bit/m^2. We also assume that the energy loss is proportional to the distance between the sender and the receiver nodes. Thus, to transmit a *k*-bit message for a distance d the energy dissipated, $E_{tx,}$ by the radio is given by equation 3.

$$E_{T_x}(k,d) = E_{elec} * k + E_{amp} * k * d^2 \tag{3}$$

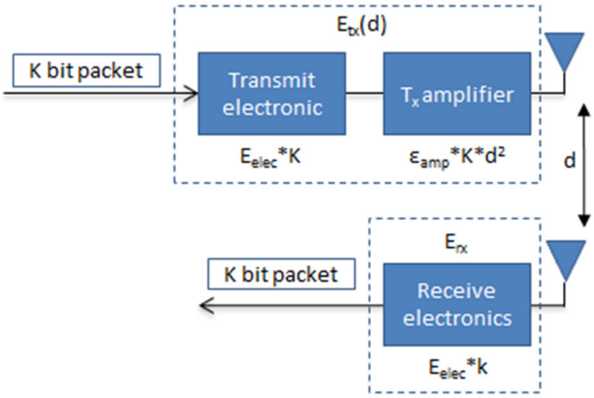

Figure 1: First order radio model (Source [15])

In our work, the multimedia content generated by a source sensor node is an adaptive H.264/MPEG4 stream [17], organized as set of Group Of Pictures (GOPs). Each GOP is a sequence of three types of frames: I-frame which can be decoded independently, P-frame which can be decoded from previous frames, and B-frames which need previous as well as succeeding frames to be decoded. We assume, similar to [18], that the size of an I frame is twice the size of a P frame and four times the size of a B-frame.

4 PROPOSED MECHANISM

In this section, we describe the design of our **E**nergy-efficient **M**echanism for Multimedia Streaming over **CO**gnitive Radio **S**ensor Networks (EMCOS). EMCOS is composed of the following building blocks: the clustering module, the cluster-head election module and the content transmission module (Figure 2).

The nodes in EMCOS are organized into a set of clusters which collaborate to deliver the multimedia content from a source node to a sink node (Figure 3). The clustering in traditional WSN has been proved to be energy efficient [16] and it consequently extends the life of the network. However, the clustering approaches adopted in WSN are not suitable for CRSN since an additional constraint, namely the spectrum availability, should be taken into consideration. In this work we propose a new clustering mechanism suitable for CRSN. First, we present a centralized approach. Then, we derive a distributed and more practical approach. Once the clusters are formed, a cluster-head is elected for each cluster to coordinate routing as well as intra-cluster and inter-cluster content transmission. In EMCOS, the clusterhead is elected in a manner which preserves the energy of the cluster and consequently of the whole network. Finally, we present the routing mechanism coordinated by the cluster-head.

Note that, in EMCOS, the intra-cluster communications are based on TDMA algorithm, coordinated by the cluster head. However, the inter-cluster communications are CSMA based due to the lack of any global coordinator and in order to avoid collisions. This strategy has been widely proved in the literature to be energy efficient in the context of WSN. In addition, in our protocol, the two mechanisms (CSMA and TDMA) can evolve in parallel, which results in low routing delay.

In the following text, we detail the clustering mechanism that we propose, and then we describe the routing/channel selection algorithm.

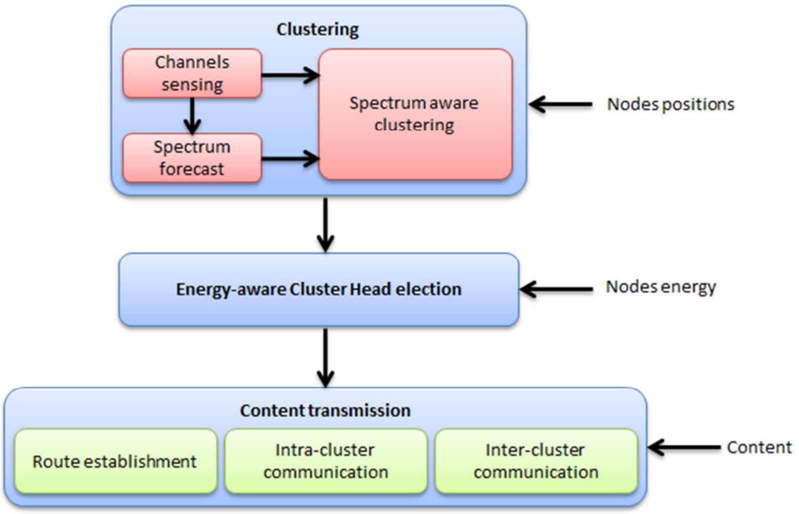

Figure 2: General architecture of EMCOS

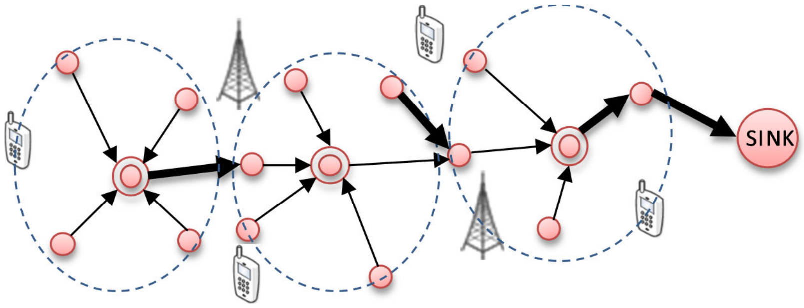

Figure 3: The network organisation

4.1 *SPECTRUM AWARE CLUSTERING*

Clustering techniques have been widely studied [19] to design effective, scalable and energy efficient delivery mechanisms for WSNs. To achieve this goal, three main questions should be answered: (a) How to build the clusters? (b) How to elect the cluster head? (c) What is the optimal number of clusters?

Authors in [18] have shown that the energy consumed in a network composed of K clusters is given by equation (5):

$$E_{network} = \sum_{k=1}^{K} E_{intra-cluster}^{k} + E_{inter-clusters} \tag{5}$$

where

$$E_{inter-clusters} = K C_0 P_r d_{max}^2 \tag{6}$$

$$E_{intra-cluster}^{k} = 2 C_0 P_r \sum_{i=1}^{N_k} d^2\left(n_i^k, clusterhead(k)\right) \tag{7}$$

and:

- C_0: constant loss factor
- P_r: minimum received power required by cluster head to correctly decode the received message.
- n_i^k: the *i*th node of the cluster *k*
- $clusterhead(k)$: cluster-head of the *k*th cluster
- $d(a,b)$ is the Euclidian distance between two points *a* and *b*
- d_{max} is the transmission range of WSN node

- N_k: the number of nodes in cluster k

According to (5), minimizing the network energy is mainly affected by the intra-cluster communication energy. This latter is closely related to the distance between the cluster-head and the cluster members (eq. 7), i.e.

$$min\ E_{network} \Leftrightarrow min\ \sum_{i=1}^{N_k} d^2\left(n_i^k, center(k)\right) \quad (8)$$

The problem of energy efficiency in WSNs has been widely tackled in the literature. However, in CRSN, the problem is different, since nodes without common channels cannot communicate with each other and consequently should be in different clusters, even if they are in the transmission range of each other. This constraint should be taken into consideration while designing a clustering mechanism for CRSN. Therefore the optimal clustering problem in CRSN can be formulated as follow:

$$\min\left(\sum_{k=1}^{K}\sum_{i=1}^{N_k} d^2\left(n_i^k, center(k)\right)\right) \quad (9)$$

Subject to:

$$\forall n_i^k, n_j^k \in Cluster\ (k)\colon |chanels\ (n_i^k) \cap channels(n_j^k)| \geq 1 \quad (10)$$

and

$$\forall n_i^k, n_j^k \in Cluster\ (k)\colon d\ (n_i^k, n_j^k) \leq d_{max} \quad (11)$$

where:

$|chanels\ (n_i^k)|$ represents the number of available channels for node(n_i^k).

The constraint (10) ensures that the cluster nodes have at least one common channel. While constraint (11) ensures that the cluster members are within the transmission range each other.

4.1.1 Constrained clustering for CRSN : Spectrum aware approach

The minimization problem presented in (9) was the subject of many clustering algorithms such as k-means, k-nearest neighbors', graph-cut, etc. The common property of these algorithms, exploited in the context of WSN, is that they try to aggregate nodes into clusters by minimizing the intra-cluster distance. These algorithms are not suitable for CRSNs since two neighbor nodes without any common channel should not be in the same cluster.

Recently, a new branch of clustering algorithms has been developed [20][21], which cluster nodes based on the distance between elements, while respecting a pairwise constraints between them: the *must-link constraint* and the *cannot-link constraint*. The *must-link* constraint on two elements n_i and n_j requires that they must be in the same cluster. While the *cannot-link* constraint between n_i and n_j requires that the two elements must be in two different clusters. In the example presented in Figure 4, we show that the pairwise constraint clustering algorithms are not relevant for CRSN. Indeed, in this example, the network is composed of three nodes: n_i, n_j and n_k having the available set of channels: (ch_1, ch_2), (ch_2, ch_3) and (ch_1, ch_3), respectively. The must constraint is "Two nodes having at least one common channel should be in the same cluster" and the cannot constraint is "Two nodes without any common channel cannot be in the same cluster". Applying the pairwise constraint clustering algorithm in this case, results into one cluster composed of all the three nodes, even when there is no common channel between the three nodes to ensure a global communication between them. In this work, we propose *spectrum group-constraint* clustering algorithm, which clusters the nodes which are geographically close and which have at least one common channel.

The clustering mechanism that we propose is based on the actual as well as the forecast of spectrum availability. Indeed, in cognitive radio networks, the availability of a channel depends on the PU activity on that channel. Forecasting the PU activity on a given channel provides a clear idea about the availability of that channel in the future. This information is exploited to build clusters of nodes having common channels in the present as well as in the future. In this way we ensure the stability of these clusters. To this end, we model the PU activity as a time series. This model is build and updated based on the regular sensing of PU activity. In the following, we describe our mechanism for PU activity forecast. We present first a centralized version of our algorithm, and then we detail the distributed version.

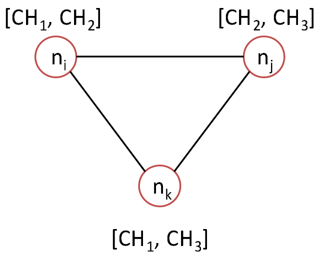

Figure 4: Pairwise constraint example

a) PU activity forecast

In order to ensure clusters stability, we not only just take into consideration the actual channel availability, but we also consider the stability of the channel over time. This is why we forecast PU activity on each channel. Indeed, a higher stability of common channels in a cluster lowers the need for re-clustering in the network. This results in lesser overhead and lower number of transmission interruptions. In the following paragraph, we detail the mechanism for PU activity estimation using time series model, adopted in our work.

In the literature [22], many PU activity models have been proposed. These models can be classified into Markovian models, models based on queuing theory and the models representing the PU activity as an ON/OFF process. In this work, we model the PU activity based on the spectrum measurements performed by RWTH Aachen University [25].First, we note that the detection of PU channel occupancy is done by a physical layer module which indicates at time t_i if PU is actually using the channel or it is in idle mode [23].

In order to study the evolution of PU activity over time we define random variables x_t and y_t as the time duration over which the PU is active and inactive, respectively, on a channel *f*. Based on the consecutive values of x_t and y_t, we construct the times series $\{x_t\}_{t\in N}, \{y_t\}_{t\in N}$ where x_i denotes the duration of the i^{th} PU activity period and y_i denotes the i^{th} PU inactivity period. N is the set of integers.

In our analysis, we use the data set obtained from the spectrum measurement study performed by RWTH Aachen University [25]. The measurements, collected from December 27th, 2006 to January 2nd, 2007 concern the 20MHz to 6 GHz bands, where most of wireless services work today. We applied the Box-and-Jenkins [26] method to analyze time series, which consists of three steps:

(1) verify the stationarity of the two time series $x_{t_{t\in N}}$ and $y_{t_{t\in N}}$

(2) Identify the time series order

(3) Estimate the model parameters

First, we run the stationarity test of the two time series $x_{t_{t\in N}}$ and $y_{t_{t\in N}}$ to verify if the studied phenomena can be modeled as time series:

$$E(z_t) = \mu \text{ Where } \mu \text{ is constant independent of instant } t \tag{12}$$

$$Cov(z_t, z_{t-j}) = \gamma^j \text{ Only depends on time lag } j \tag{13}$$

Where $z_{t_{t\in N}}$represents the studied time series $x_{t_{t\in N}}$ or $y_{t_{t\in N}}$.

The obtained results showed that the two stationarity conditions (12) and (13) are satisfied in all the time series resulting from the data. Consequently x_t and $y_t$can be analyzed using Box and Jenkins method and can be modeled as Auto-Regressive Moving-average (ARMA) time series. For lack of space, we do not provide the details related to the Box and Jenkins steps for the times series $x_{t_{t\in N}}$. The same process is applied to the series $y_{t_{t\in N}}$ to determine its model parameters. A detailed study is provided in our previous work [27].

In order to analyze the resulting time series, we used Minitab [28]. The obtained results indicate that the best appropriate model is the ARMA (3,1). Based on this, the PU occupancy model can be written as:

$$x_i = \mu + \varphi_1 x_{i-1} + \varphi_2 x_{i-2} + \varphi_3 x_{i-3} + \theta_1 \varepsilon_{i-1} + \varepsilon_i \tag{14}$$

Where μ represents the mean of x_i. $\varphi_1, \varphi_2, \varphi_3 \wedge \theta_1$ denote the PU occupancy model parameters ($\varphi_1, \varphi_2, \varphi_3$ related to the AR part and θ_1 related to the MA part). We applied the same test to the PU inactivity time $y_{t_{t \in N}}$ and we conclude that it follows an ARMA (2, 1) model.

After identifying the order of the ViCoV PU occupancy model, the next step is to estimate its parameters. For the autoregressive part (AR), the parameters can be computed using the Yule-Walker equations [34]:

$$\gamma_m = \sum_{k=1}^{p} \varphi_k \, \gamma_{m-k} + \sigma_\varepsilon^2 \delta_m \tag{15}$$

Where m = 1,…,*p* yielding (*p*+1) equations. γ_m is the auto-correlation of Y,$\sigma_\varepsilon$is the standard-deviation of the input noise process, and the δ_m is the Kronecker Delta function. Using Equation (15) AR (p) parameters can be estimated by replacing the covariance with its estimated values. The estimated AR parameters are then used to deduct the MA parameter by identification.

b) Centralized clustering algorithm

In the centralized version of our mechanism that we call CEMCOS (centralized EMCOS), we assume a centralized sink node which receives data from all nodes, mainly: the geographic position (GPS coordinate), the available channels and the residual energy. The sink node, based on the collected data, runs the centralized clustering algorithm and informs the different nodes about the clustering result.

After a sensing phase, each node builds a list of available channels, and statistically predicts the availability of each channel over time using the time series model presented in the previous sub-section. Let A_i (*t*) be the channels availability matrix of node *i* at time *t*, where a_i^c is a Boolean variable which indicate if the channel *c* is available for node *i*.

$$A_i(t) = [a_i^0, a_i^1, \dots, a_i^{C-1}]^{\mathrm{T}}$$

Similarly, we define the expected channels time availability matrix E_i(t) where e_i^c is the expected time availability of the channel c in the node *i*.

$$E_i(t) = [e_i^0, e_i^1, \dots, e_i^{C-1}]^{\mathrm{T}}$$

The clustering algorithm that we propose is inspired from the complete-link hierarchical agglomerative clustering algorithm [21]. This algorithm first considers each node as a cluster. Then, it regroups the nearest clusters which respect the pairwise constraint. The process is repeated for the obtained clusters until the stop condition is satisfied. In our proposed algorithm, we replace the pairwise constraint with the spectrum group-constraint, i.e. we regroup the geographically nearest clusters which satisfy the condition "all the nodes should have at least one common channel". Moreover, we promote clustering of nodes having more common channels which in turn have high expected availability time. For that purpose, we define the spectrum affinity coefficient between clusters *k* and *k'* as follows:

$$SA(k,k') = \frac{\sum_{c=0}^{c=C}((a_k^c * e_k^c) * (a_{k'}^c * e_{k'}^c))}{d(k,k')} \tag{16}$$

where:

$$d(k,k') = \max_{n_i \epsilon k, n_j \epsilon \acute{k}} d(n_i, n_j) \tag{17}$$

For a cluster *k*, we redefine the variables:

a_k^c : Boolean variable which indicates if the channel *c* is a common channel for all cluster *k'* members

e_k^c : Mean available time of the common channel *c*

We define, then, the clusters spectrum affinity matrix AM as follow:

$$\begin{array}{c|cccc} & \boldsymbol{k_0} & \boldsymbol{k_1} & \cdots & \boldsymbol{k_l} \\ \hline \boldsymbol{k_0} & 0 & \mathrm{SA}(k_0,k_1) & \cdots & \mathrm{SA}(k_0,k_1) \\ \boldsymbol{k_1} & \mathrm{SA}(k_1,k_0) & 0 & \cdots & \mathrm{SA}(k_1,k_l) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{k_l} & \mathrm{SA}(k_l,k_0) & \mathrm{SA}(k_l,k_1) & \cdots & 0 \end{array}$$

The centralized clustering algorithm that we propose is described in *Algorithm* 1. The sink node, after receiving data from all nodes, in the data-collection step, initializes each node as a cluster. It also computes the spectrum affinity matrix (AM). For each line in the matrix (corresponding to a cluster k_i), it selects another cluster corresponding to a high SA value (k_j). The two clusters k_i and k_j are deleted from the AM and they are replaced by a new cluster k_l. The spectrum affinity coefficients of cluster k_l are computed using (16), where $a_{k_l}^c = a_i^c * a_j^c$ and $e_{k_l}^c = min\left(e_i^c, e_j^c\right)$. A new round of merge and replacement is started for the clusters, and the iterative process stops if the maximum number of clusters is reached or if there is no new merging possibility between any two clusters.

Algorithm 1: CEMCOS algorithm

```
Step1: Receive nodes data : position, available channels $A_i(t)$, expected channels time availability $E_i(t)$

Step2: Spectrum group-constraint clustering algorithm
    (1) Initialize each node as cluster
    (2) Compute the spectrum affinity coefficient for each pair of clusters and build the affinity matrix
    (3) For each cluster $k_i$ find, in the AM, the cluster $k_j$ with high affinity coefficient
    (4) Merge the two clusters $k_i$ and $k_j$ and remove the corresponding lines and columns in the affinity Matrix
    (5) Add a new cluster $k_l$ : $a_{k_l}^c = a_i^c * a_j^c$ and $e_{k_l}^c = \min(e_i^c, e_j^c)$ and add it in the Spectrum Affinity matrix
    (6) If AM <> (0) or number of clusters = $K_{max}$ Go to sub-step (3) // (0) is the zero matrix, i.e. there is no possibility of cluster the network

Step3: Inform nodes by the clustering results
```

c) distributed clustering algorithm

Whereas the centralized algorithm CESMOS allows an efficient spectrum aware clustering of the network, it suffers from some drawbacks. The signaling messages, exchanged between the nodes and the sink, generate extra traffic which overloads the network. Furthermore, the complexity of the algorithm is related to the square of the matrix size, and consequently with the size of the network. The complexity of the proposed algorithm is $O(n^2)$, where n is the size of the network. To deal with these limitations, we propose the distributed version of ESMOS, namely DESMOS (Distributed ESMOS).

The main idea behind DESMOS remains similar to CESMOS. Initially, each node is considered as a clusterhead of a single-member cluster. Based on the spectrum affinity between the network nodes, the clusters are built in an incremental way. However, rather than a centralized entity performing the clustering process, each node broadcasts its available channels and their expected availability time to its neighborhood. A cluster, represented by its clusterhead, decides to join another neighbor cluster based on the spectrum affinity coefficient between the two clusters. After the building of the clusters, each node periodically senses the common channels of its cluster. If a node detects that it is disconnected from the cluster (no common channel is available, due to PU activity), the node considers itself as a cluster head of a new single-member cluster and a new beaconing-merging process is started. A pseudo-code of the proposed DESMOS algorithm is provided in *Algorithm* 2.

Two kinds of beacons are used in the network: *cluster_head_beacon* and *node_beacon*. The cluster head periodically broadcasts a *cluster_head_beacon***,** to inform about the *cluster_id*, the cluster members, the common channels and their expected availability time. A cluster head receiving *cluster_head_beacons***,** computes the affinity coefficient with each cluster and selects the cluster with the highest coefficient. It sends a *merge_request* message to the selected cluster.

A cluster head receiving a *merge_request* message, checks if the requesting cluster is the closest. If it is the case, the two clusters merges. A new cluster head is elected and a *cluster_head_beacon* message is broadcasted in the network. Each node of the network periodically broadcasts a standard *node_beacon***,** to mainly inform about its residual energy, the available channels and the channels' expected availability. The cluster head accordingly updates the common channels and broadcasts *cluster_head_beacon* if any changes are observed.

Algorithm 2: DESMOS pseudo-algorithm (executed by each network node *i*)

```
Periodic_sensing_beaconing () // executed periodically by each node
    Sensing () // Sens channels to check if there is always common available channels with the cluster
    if (cluster_id = Null) or (available_commun_channels (cluster_id) = 0)
        Cluster_id = new cluster_id() // new cluster
        Cluster_members = {$i$} // a new cluster composed of one node
        Cluster_head_id = node_id // the node is the cluster head of the new cluster
    If  is_cluster_head ($i$)
        Broadcast_cluster_head_beacon (Node_id, Cluster_id , cluster_members, $A_{ki}$, $E_{ki}$)
    Broadcast_beacon (node_id, cluster_id, position,  residual_energy, $A_i$, $E_i$)  // beacon of an ordinary node

OnEvent: receive_cluster_head_beacon (node $j$)
    If  is_cluster_head($i$) // Executed by the clusterhead
        SA = compute_SA ($i$, $j$) // Compute the spectrum affinity between cluster $k_i$ and cluster $k_j$
        $K_l$ = Select_cluster_with_max_SA() // among the cluster head requests, select the closest one
        Send_merge_request ($K_l$)

OnEvent: receive_merge_request (cluster $K_j$)
    If  is_cluster_head($i$) // Executed by the clusterhead
        SA = compute_SA ($i$, $j$) // Compute the spectrum affinity between cluster $k_i$ and cluster $k_j$
        If ($K_j$ = Select_cluster_with_max_SA()) // if the requester cluster is the closest one then merge
            Cluster_id = new cluster_id() // New cluster
            Cluster_size = cluster_size($k_i$) + cluster_size($k_j$)
            Cluster_members = cluster_members ($k_i$) ∪ cluster_members ($k_j$)
            Cluster_head_id = elect_cluster_head ($k_i \cup k_j$)
            Broadcast (Node_id, Cluster_id , cluster_members, $A_{ki}$, $E_{ki}$) // cluster head beacon

OnEvent: receive_node_beacon (node $j$)
    If is_cluster_head($i$) // Executed by the clusterhead
        Update ($A_{ki}$)
        Update ($E_{ki}$)
        If common_channels_changes
            Broadcast_cluster_head_beacon (Node_id, Cluster_id , cluster_members, $A_{ki}$, $E_{ki}$)
```

4.2 ENERGY AWARE CLUSTER HEAD ELECTION

In clustered WSNs, the cluster head plays a major role in content routing from the source to the sink. Each cluster member sends the data to its cluster head, which forwards it to the sink node, directly if it is in the transmission range, otherwise via other intermediate cluster heads. Selecting the cluster head is then of extreme importance to ensure the continuity of the service. Most of works in the literature select the node with highest residual energy as the cluster head. However, this strategy does not ensure maximum prolongation of the network life. Indeed, if the node with the highest energy is situated on the geographic boundary of the cluster, the cluster members will spend higher energy to reach the cluster head, which is not an optimal energy utilization of the network. This is why in DEMCOS we propose a new cluster head election mechanism which ensures efficient energy utilization of the cluster nodes and consequently longer life of the network. Rather than electing a single cluster head always, each node in the cluster is elected for a certain period, depending on its residual energy and other cluster members' energy consumption. In this way, the cluster head function rotates between all the cluster members.

The energy consumed by cluster head is related to the data received from the cluster members and the data transmitted / received from the other cluster heads to reach the sink. Whereas, the energy consumed by a cluster member is related to the data transmitted to the cluster head. In this work, we assume that the energy consumed for signaling is neglected, since the size of the signaling messages is negligible as compared to the size of the data (multimedia content).

Let's assume e_i^j the energy consumed by a node *i* per time unit to transmit data to the cluster head j, r_j represents the residual energy of the node j, and x_j represents the times that node j is a cluster head. The energy consumption of a cluster composed of *n* nodes can be modeled as follows:

$$E * X = R \tag{18}$$

Where

$$E = \begin{pmatrix} e_1^1 & e_1^2 & \cdots & e_1^n \\ e_2^1 & e_2^2 & \cdots & e_2^n \\ \vdots & \vdots & & \vdots \\ e_n^1 & e_n^2 & \dots & e_n^n \end{pmatrix}$$

$$X = [x_1, x_2, \dots, x_n]^{\mathrm{T}}$$

$$R = [r_1, r_2, \dots, r_n]^{\mathrm{T}}$$

The variables in this system are the x_i. Resolving this equation is equivalent to defining the time that each node of the network should be the cluster head.

Initially, once the clusters are built, the node with the highest residual energy is elected as the temporary cluster head. It receives the information about the position and residual energy of cluster nodes. The cluster head then computes the variables e_i^j and resolves the equation (18) using the Gauss–Jordan elimination algorithm presented in *Algorithm* 3. The temporary cluster head informs the other nodes about the schedule according to which each node will play the role of the cluster head (x_i). The cluster head also informs about the channel to be used based on the expected availability time of the common channels.

We note that the cluster-head is elected periodically, depending on the residual energy of the cluster nodes. This means that the cluster-head function rotates among different cluster nodes, which is also valid for the last cluster on the way towards the sink, which we call as the last mile cluster. Consequently, there is no hotspot problem (the problem of last mile node being overburdened) in the last mile cluster, because the cluster-head function rotates among the different nodes of this cluster, even if the sink is static.

Algorithm 3: The Gaussian elimination algorithm for CRSN cluster head election

```
Input:  A = (E*X R)
Output: X = A

for k = 1 ... n:
   i_max  := argmax (i = k ... m, abs(A[i, k]))
   Swap rows(k, i_max)
   for i = k + 1 ... n:
     for j = k + 1 ... n:
       A[i, j]  := A[i, j] - A[k, j] * (A[i, k] / A[k, k])
     A[i, k]  := 0
```

4.3 *ROUTING MECHANISM*

The route establishment in DEMCOS is performed in a proactive manner. The overall overhead is minimized because only the cluster heads are involved in this process. The routing of the content from the source node to the sink consists of three main phases. Initially, a route is established from the source cluster to the sink node. Then, the members of different clusters send their content to the cluster head only if they do not have a valid or optimal route to the sink. This is done because if the optimal route passes through a given node, then it will be non-optimal to send the data to the cluster-head which will only resend it to the node in order to be forwarded to the next cluster.

Finally, inter cluster transmission is used to route the content from a cluster to another till it reaches the sink. We describe the each routing phase in the following text:

a) Route discovery

At the end of this step a route is established from a cluster K_i to the sink node. To achieve this goal, messages of discovery and reply are exchanged between the cluster head and its cluster members as well between the members of different clusters. We note that the road discovery is inspired from the CR AODV [30]. We describe the different steps in the following:

Route request step: In order to establish the route from a cluster K_i to the sink, the cluster head of K_i (CH_i), broadcasts a route request R_REQUEST message on the CC channel. A member of K_i receiving the message might have the requested route in its cache. In this case, it broadcasts R_REP messages. Otherwise, it waits for a random backoff time, and rebroadcasts the R_REQUEST message further.

Route announce step: A member n_j of a neighbor cluster K_j, receiving the R_REQUEST, answers with a R_REP message if it knows the route to the sink, otherwise it forwards the R_REQUEST to its cluster head CH_j. If CH_j, has the route in its cache, it broadcasts a R_REP message. In this case, all cluster K_j members save/update the route to the sink in their caches, and only the node n_j rebroadcasts the R_REP message. If the cluster head does not know the route to the sink, then it repeats the route request step.

We note that, in order to reduce the signaling overhead, a node n_k receiving a R_REQUEST reacts only if it receives this message for the first time, otherwise it ignores the message.

All the members of cluster K_i receiving the R_REP message save/update the route to the sink, and only the node n_i forwards the message to the cluster head on CC channel.

b) Intra cluster communication

Once the route to the sink is established, nodes in the cluster send their content to the cluster head, which forwards it hop by hop to the sink. The intra cluster communication is TDMA based and is coordinated by the cluster head. The protocol is described in the following paragraphs.

TDMA slots reservation step: The cluster head announces the request phase, which operates in CSMA mode on the CC channel. If the cluster member has a valid route to the sink, it sends its content directly via inter cluster communication using the learned route. Otherwise, the node transmits its content via the cluster head. In this case, the node sends D_REQUEST message to the cluster head, announcing the number of requested time slots, the QoS level of the content (real-time, non-real-time) and the route to the sink if available.

TDMA slots allocation and transmission step: On receiving D_REQUEST messages, the cluster head allocates the TDMA time slots for each node, and schedules the transmission channels, based on the expected availability time of each cluster channel using the time series forecast model presented in 4.1.1. The cluster head broadcasts T_SCHED message to inform cluster members about the allocation of TDMA time slots as well as the scheduling of the channels. On receiving this message, the cluster nodes synchronize with the cluster head and start transmitting their content using the allocated time slots and the designated channel.

We note that in case of high TDMA slots request, the cluster head prioritizes the I frames. In such context, the cluster head fulfill first all the TDMA slots request related to I-frames. Then, the remain TDMA slots are allocated to the P and the B frames.

c) Inter cluster communication

Inter cluster communication is needed to reach the sink. For that purpose, channel synchronization is required between the egress node of the upstream cluster and the ingress node of downstream cluster. Two inter cluster communication situations are possible. A node can have a content to transmit and also have a valid route. The second situation is when a node receives content from the cluster head to forward it to the downstream cluster. In both cases the edge node should synchronize with a node from the downstream cluster.

In DEMCOS, content transmission from a node to the cluster head or vice versa is realized in TDMA mode. However, a cluster member outside its TDMA slots is tuned to CC channel and works in CSMA mode. An egress node n_i of cluster k_i having a content to send/forward to the downstream cluster k_j, broadcasts *C_RTS* message on CC channel, in which it announces its available channels and the destination cluster k_j. A node n_j

from the downstream cluster k_j, receiving the messages and having at least one common channel with node n_i, responses with *C_CTS* on CC channel, announcing its decision about the communication channel. The node n_i then sends its content and waits for acknowledgement. If downstream cluster node n_j does not acknowledge the packet, the upstream cluster node n_i rebroadcasts a *C_RTS* message and the process is repeated until the content is acknowledged or a maximum transmission tentative is reached, and the packed in through.

5 PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed mechanism DEMCOS, using the network simulator NS-2. We compare our approach with SEARCH [14] and SCEEM [18]. The performance of the three protocols is compared in terms of Peak Signal-to-Noise Ratio (PSNR), frames loss, frames delay and energy consumption.

The simulations were performed using NS-2 version 2.31[31], extended to support cognitive radio in the physical layer using the (CRCN) patch [32][24].

The NS-2 CRCN patch does not implement the PU activity. We developed the PU activity module, which generates and maintains track of the PU activity on each band.

To implement our solution in NS-2, we considered the simple Mac protocol *Maccon.cc* provided in CRCN patch, which is multi-channel Mac protocol. In this protocol the channel selection is made at the Mac layer. We modified it in such a way to be driven by our protocol DEMCOS, which in turn is implemented as a routing agent in NS-2.

In all the experiments, we randomly deploy PUs and SUs nodes in a 500m * 500m area. The sink is chosen randomly for each simulation run. The PU occupies the channel randomly, and the SU can access the channel if no PU is active on this channel. The protection range for PU is 20 m, which mean the PU's neighbors within this range cannot access this channel. The radio propagation model adopted in our simulation is two-ray ground model implemented in NS2.

Regarding the transmitted video, we adopt the widely used "akiyo cif" video, composed of 300 frames at a resolution of 360 x 486. The frames are packed up in 560 packets of 1024 bytes each. In order to evaluate the PSNR metric of the received video, we use the video quality evaluation tool-set EvalVid [33] on the reconstructed raw videos. We note that the obtained results are the mean of 30 independent simulation runs for each scenario, represented at a confidence interval of 95%. In Table 4, we summarize the simulation parameters.

Parameter	**Value**
Area	500m * 500m
Number of PU	[2, 20]
Number of SU	150
Number of channels	[2, 18]
Sensing time per channel	5ms
Channel switching time	80µs

Table 1: simulation parameters

5.1 PSNR

We first evaluate the performance of our approach DEMCOS in terms of video quality. We compare the received video quality at the sink node for the three protocols: DEMCOS, SCEEM and SEARCH, under different spectrum and streaming conditions. Note that a PSNR value higher than or equal to 30 for a video is considered as a good enough quality.

Figure 5 shows the average Peak Signal-to-Noise Ratio (PSNR) of the reconstructed video at the sink node under different spectrum availability conditions. The number of PUs in this scenario varies from 2 to 20. The general trend is that the video quality decreases with the increasing number of PUs in the network. Indeed, increasing PU activity in the network leads to less available channels and consequently more packet loss. We note that the PSNR is very low when SEARCH is used, mainly when the number of PUs is high. This is explained by the fact that SEARCH is a reactive protocol: If the PU activity is detected on the current used channel, a new route is calculated. When the number of PUs increases, the frequency of route recalculation increases and consequently more packets are lost. This results in a low PSNR. On the other hand, the video quality with DEMCOS is better, since the protocol is based on the PU activity forecast. Indeed for the intra-cluster as well as the inter-cluster communications, DEMCOS selects a stable channel which is estimated to have lower PU activity. In addition, in DEMCOS, the nodes are clustered based on the number of common channels for improved stability. If a PU becomes active on a channel, the cluster-head switches to another common channel to communicate with the cluster members, without the need to a new route discovery. In addition, we note that the video quality with SCEEM is worse than with DEMCOS, since the communication channel in SCEEM is selected without any estimation of PUs activity on the channel. Thus SCEEM results in more channel switch and more re-clustering of the network, which leads to more packets lost and lower PSNR as compared to our approach DEMCOS.

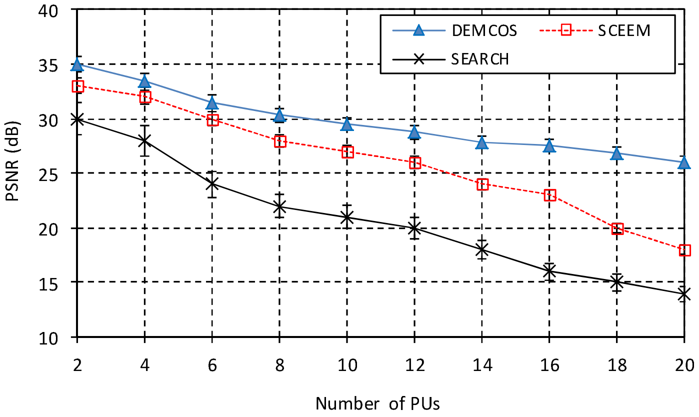

Figure 5: PSNR Vs. Number of PUs

In second scenario, we measured the PSNR of the received videos in the sink node while varying the number of video sources from 2 to 20. The number of PUs is fixed to 10 and the number of channels is also fixed to 10. Figure 6 shows that video quality with DEMCOS decreases when the number of video sources increases. This is explained by the fact that when the number of video sources increases, the cluster head cannot fulfill all the demands for TDMA slots reservation. In this case the video source should decrease the video rate as explained in section 3. The protocol SCEEM adopts the same strategy, however our approach DEMCOS outperforms SCEEM because it selects the appropriate channels for clustering and for content transmission, in order to avoid channel switching and re-clustering.

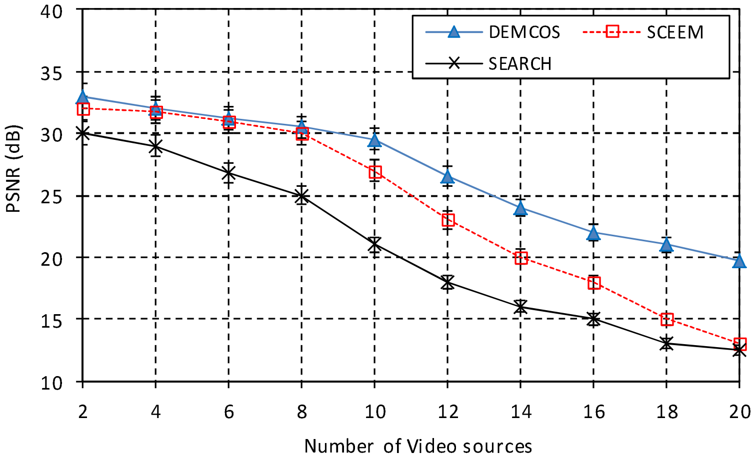

Figure 6: PSNR Vs. Number of video sources

We now fix the number of video sources to 10, the number of PUs to 10 and we vary the number of channels from 2 to 16. The results are presented in Figure 7. The graph can be subdivided to two main parts. The first part corresponds to the case when the number of channels is between 2 and 10. We note that in this part the video quality increases exponentially with the number of channels. With DEMCOS, the PSNR increases by 5 dBs when two additional channels are available (from 6 to 8 channels), and by three dBs with SCEEM. The difference in video quality between our approach and the two other approaches is more significant in case of channels scarcity (<10). For 8 channels, the quality with DEMCOS is almost acceptable (~30dB), however the videos quality with SCEEM, for the same number of channels, have an average PSNR of 20 dB, and 15 dB for SEARCH. This shows that our approach exploits the available channels better, mainly in case of channels scarcity, thanks to the new clustering and the channel scheduling mechanism, which takes the PU activity trend as well as the common channels into consideration. In the second part of the graph ([10-18]), we note that the increase in the number of available channels has less effect on the video quality. Indeed two additional channels (14 to 16) enhance the video quality by 1 dB for DEMCOS and SEARCH.

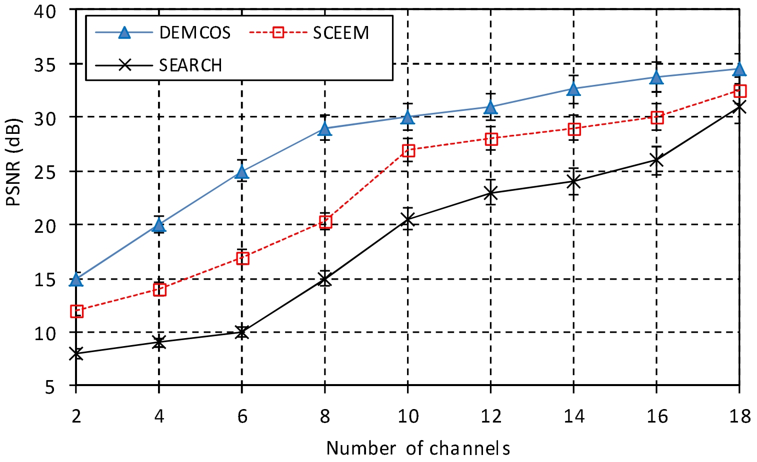

Figure 7: PSNR Vs. Number of channels

5.2 *FRAMES DELAY*

We now evaluate the three protocols in terms of average end-to-end delay under different PUs densities and for different video source bit rate (300 Kbps and 600 Kbps). We consider a scenario where the number of channels is set to 10 and the number of video sources is also set to 10. The results are represented in Figure 8. All the three mechanisms achieve lower delays when the PU activity is low, since there are more transmission opportunities without any interferences and retransmissions. We also note that when the video bitrate is high , the delay is naturally lower as compared to when the video bit rate is low, for all the three mechanisms. However, with SEARCH we obtain the highest average delay among the three mechanisms. Indeed, when the channel availability changes frequently due to higher PU activity, the route discovery mechanism is frequently triggered , which in turn results into a long end to end packet delay. In addition, SEARCH involves each and every node in the route/channel discovery mechanism. This results in high overhead and consequently higher delay. On the other hand, in our approach, the transmission channels are selected based on the estimation of the PU activity on the channel. Hence, low number of channel switching and low number of re-clustering are performed. This explains the low delay in DEMCOS as compared to SCEEM. The delay with our approach is 500% lower than in SEARCH and 200% lower than SCEEM.

The results in Figure 9 confirm the impact of the video bitrate on the end to end delay. The difference in terms of delay is clearly visible when the bitrate is low (between 100 Kbps and 200 Kbps). However, the difference is lower at higher bit rates (between 600 Kbps and 700 Kbps), because the bit rate is almost within the TDMA capacity. The delay continues to grow because of the retransmissions due to the PU activity on the channels.

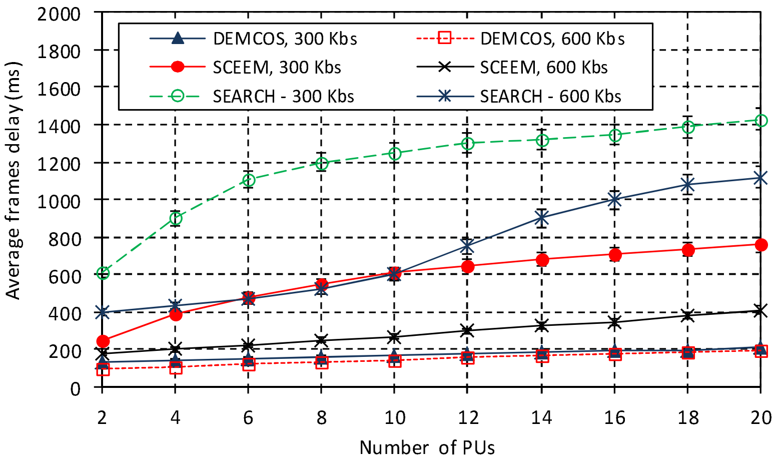

Figure 8: Frames delay Vs. Number of PUs

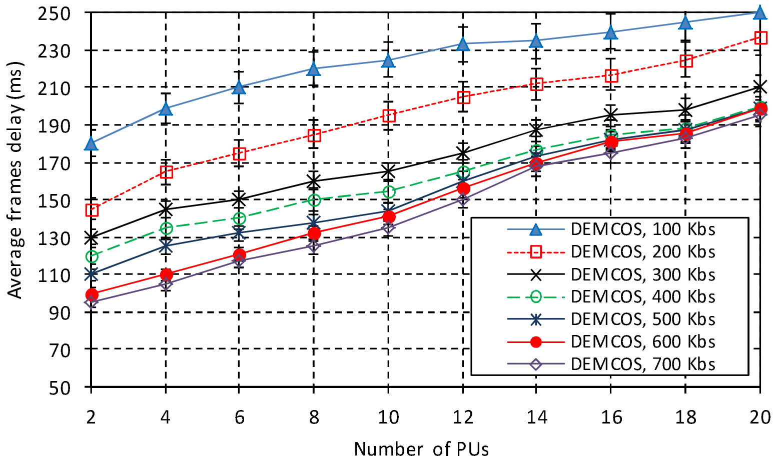

Figure 9: Frames delay Vs. Number of PUs

We now study the impact of the number of clusters on the end to end delay in DEMCOS and SCEEM. We adopt the same previous scenario, and we fix the bitrate to 400 kbps, the number of PU is 10 and the number of channels is also fixed to 10. We modified our clustering algorithm and that of the SCEEM to fix the number of clusters in advance. The results are represented in Figure 10. We note that the frame delay increases rapidly with the number of clusters. It goes from 400 ms with SCEEM for 5 clusters to about 900 ms when the number of clusters is 30, under bitrate of 300 Kbps. Similarly, the frame delay goes from about 100 ms to more than 300 ms in the case of DEMCOS under 300 Kbps.

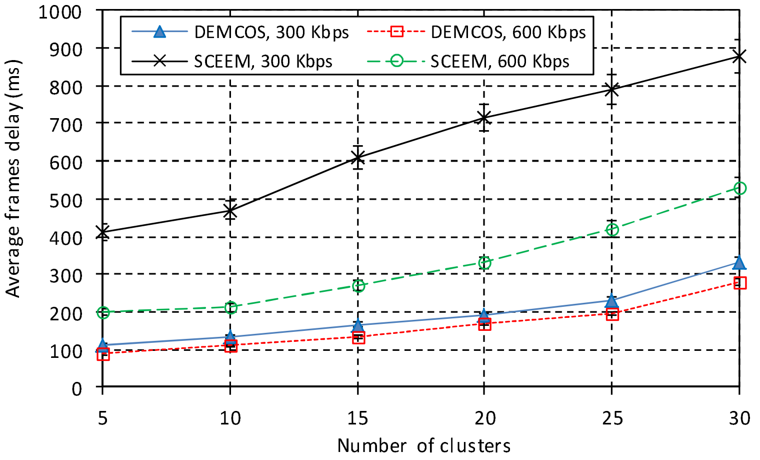

Figure 10: Frames delay Vs. Number of clusters

5.3 ENERGY CONSUMPTION

We study the energy consumption of our solution and derive the optimal number of clusters, from the point of view of energy efficiency. We vary the number of clusters from 4 to 16 and we measure the average energy consumed by a node. The obtained results are represented in Figure 11.

Firstly, we note that DEMCOS saves 50% more energy as compared to the SCEEM protocol. This is essentially due to the cluster head mechanism proposed in our proposal. Indeed, in DEMCOS the cluster head is elected periodically in such a way to optimize the overall energy consumption of the cluster in each round. Whereas, in SCEEM, the node with higher energy is always elected as the cluster head, even if it is not the optimal one. Secondly, we can see in the graph that the optimal number of clusters in these conditions is 12. When the number of clusters is very low (4 clusters), the average distance between the cluster head and the cluster members is longer. This leads to nodes having to spend more energy to reach the cluster head. When the number of clusters increase (from 4 to 12), the cluster head is closer to the cluster members, and the average energy consumed by a node is lower when the number of clusters is 12. When the number of cluster is bigger than 12, we note that the energy consumption starts to increase. This is because more clusters mean more nodes are involved in the forwarding of the content and consequently more energy is consumed.

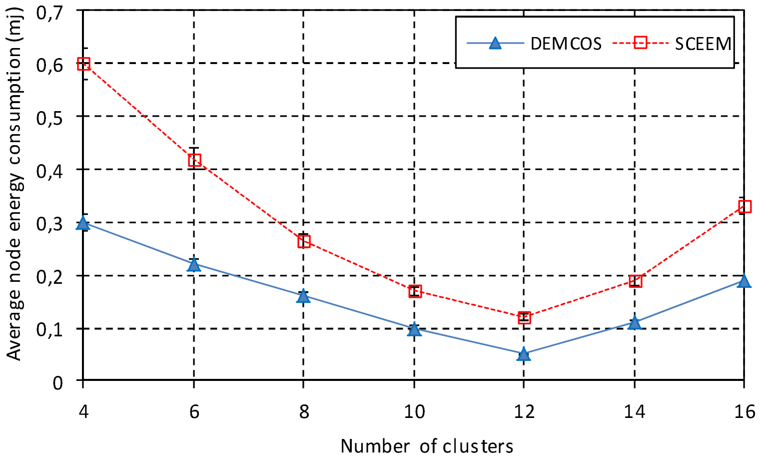

Figure 11: Average node energy consumption Vs. Number of clusters

We now fix the number of clusters to 12, we vary the number of PUs, and we compare the average energy consumption for the three mechanisms: DEMCOS, SCEEM and SEARCH (there is no clustering in the case of SEARCH). We note that cluster based mechanisms (DEMCOS and SCEEM) outperform the protocol that does not use clustering. This confirms the general effectiveness of clustering to save energy in WSN. Moreover, we note a slight increase in energy consumption for SEARCH, when the number of PUs increases. We explain this by the fact that more route re-calculations are needed with increasing PUs activity. Consequently more energy is consumed.

On the other hand, we note a similar phenomenon when varying the number of clusters for DEMCOS and SCEEM. The consumed energy decreases when the number of PUs increases for PUs < 10. However, the consumed energy increases when the number of PUs increases more than 10. We explain this by the relationship between the number of PUs and the number of clusters. In general, more PUs in the network lead to more clusters. This is why we obtain a similar shape of the performance curve when increasing the number of clusters (Figure 12).

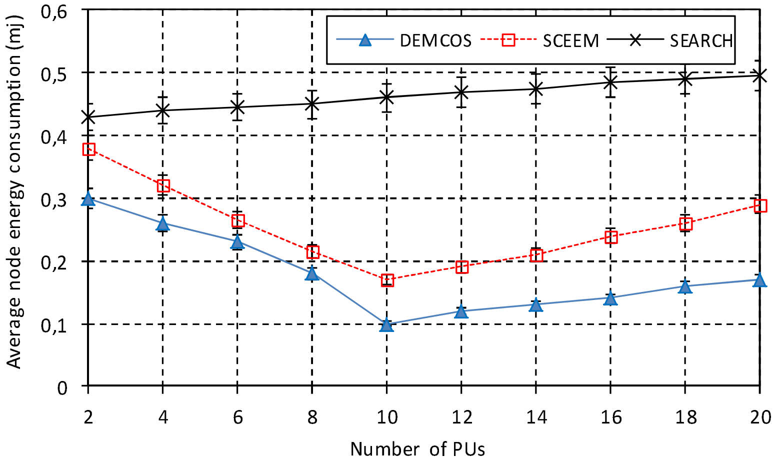

Figure 12: Average node energy consumption Vs. Number of PUs

5.4 FRAMES LOSS

We also look at the performance in terms of how many video frames were correctly delivered. We measured the overall frames delivered for the three protocols. We also measured the delivery of different types of video frames: I, P and B. The results are represented in Figure 13. We can clearly see the effectiveness of our prioritization mechanism for the I-frames. Remember that I frames are the most important video frames because other frames use them for reference during video encoding.. Indeed about 95% of I-frames are correctly received, while only 80%of B – frames is correctly received due to the differential priority provided to these frames.

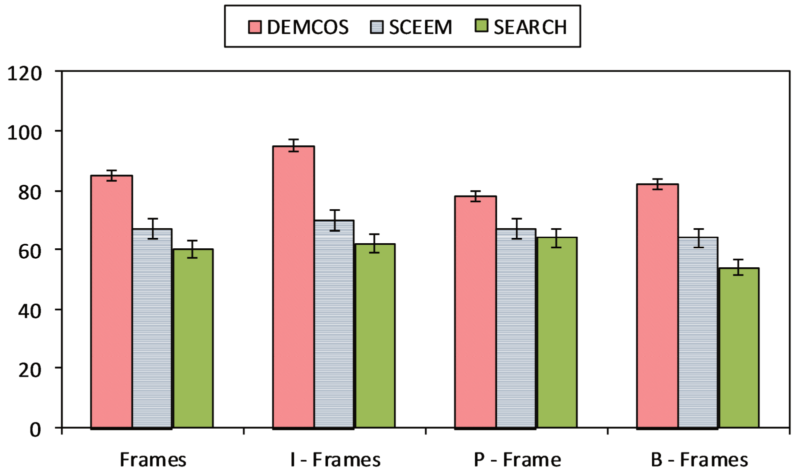

Figure 13: Average video frames delivery

We conclude from this performance study, that DEMCOS is a valid solution for video streaming over CRSN and it widely outperforms the pioneering state of the art protocols. The results showed that DEMCOS can ensure good video quality. DEMCOS especially performs well when the ratio of PUs and SUs is less than 1/15, the number of channels is more than 10 and when the ratio of number of clusters to number of nodes is around 12/150. Indeed, under these conditions, the video has a PSNR greater than 30, average frame delay of 160 ms and very low energy consumption.

6 CONCLUSION

In this paper, we proposed a video streaming solution called EMCOS for CRSN, which operates under varied spectrum conditions while minimizing the energy consumption in the network. Additionally, we outlined a scenario of video surveillance where EMCOS is a perfect solution for multimedia streaming over cognitive radio wireless sensor network, which requires high bandwidth, low delay and high PSNR. EMCOS clusters the multimedia sensor nodes based on the spectrum availability as well as the spectrum forecast in addition to node positions. This is done to ensure the stability of the clusters. In addition, EMCOS elects the best cluster heads to ensure high energy utilization in the network. Moreover, it selects the best channels which are free of PU activity and which will stay free for the longest period.

Using simulations, we showed that as compared to two related existing mechanisms (SEARCH, SCEEM), EMCOS allows to significantly improving the received video quality in terms of delay, PSNR and frames loss and optimize the energy consumption. Indeed, EMCOS increases the video PSNR by 50% and reduces the average frame delay by more than 40%, while reducing the energy consumption by 35%, as compared to the SCEEM protocol.

References

[1] WANG, Beibei et LIU, KJ Ray. Advances in cognitive radio networks: A survey. Selected Topics in Signal Processing, IEEE Journal of, 2011, vol. 5, no 1, p. 5-23.

[2] Akyildiz IF, Lee W-Y, Chowdhury KR: CRAHNs: Cognitive radio ad hoc networks. Ad Hoc Networks 2009,7(5):810–836.

[3] Jha SC, Rashid MM, Bhargava VK, Despins C: Medium access control in distributed cognitive radio networks. Wireless Commun 2011,18(4):41–51.

[4] Jha SC, Phuyal U, Rashid MM, Bhargava VK: Design of OMC-MAC: an opportunistic multi-channel MAC with QoS provisioning for distributed cognitive radio networks. IEEE Trans. Wireless Commun 2011,10(10):3414–3425.

[5] Ehsan, Samina, and Bechir Hamdaoui. "A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks." Communications Surveys & Tutorials, IEEE 14.2 (2012): 265-278.

[6] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, and Jean Marie Bonnin. "Wireless sensor networks: a survey on recent developments and potential synergies." The Journal of Supercomputing (2013), Springer

[7] Bicen, A. Ozan, V. Cagri Gungor, and Ozgur B. Akan. "Delay-sensitive and multimedia communication in cognitive radio sensor networks." Ad Hoc Networks 10.5 (2012): 816-830.

[8] Rehmani, Mubashir Husain, Stéphane Lohier, and Abderrezak Rachedi. "Channel bonding in cognitive radio wireless sensor networks." Mobile and Wireless Networking (iCOST), 2012 International Conference on Selected Topics in. IEEE, 2012.

[9] Liang, Zhongliang, and Dongmei Zhao. "Quality of service performance of a cognitive radio sensor network." Communications (ICC), 2010 IEEE international conference on. IEEE, 2010.

[10] Liang, Zhongliang, et al. "Delay performance analysis for supporting real-time traffic in a cognitive radio sensor network." Wireless Communications, IEEE Transactions on 10.1 (2011): 325-335.

[11] G. H. E. Fard, M. H. Yaghmaee, R. Monsefi, “An Adaptive Cross-Layer Multichannel QoS-MAC Protocol for Cluster Based Wireless Multimedia Sensor networks,", Proc. of Ultra Modern Telecommunications & Workshops (ICUMT’09), pp.1-6, 12-14 Oct. 2009.

[12] Y.-F. Wen, W. Liao, “On QoS Routing in Wireless Ad-Hoc Cognitive Radio Networks," Proc. of 71st IEEE Vehicular Technology Conference (VTC’10), pp.1-5, 16-19 May 2010.

[13] L. Zhou, X. Wang, W. Tu, G. Mutean, and B. Geller, “Distributed Scheduling Scheme for Video Streaming over Multi-Channel MultiRadio Multi-Hop Wireless Networks," IEEE Journal on Selected Areas in Communications, vol. 28, no. 3, pp. 409-419, Apr. 2010.

[14] K.R. Chowdhury, M.D. Felice, “SEARCH: A routing protocol for mobile cognitive radio ad-hoc networks,", Computer Communications, vol. 32 pp. 1983-1997, 2009.

[15] Digham, F.F.; Alouini, M.S.; Simon, M.K. On the energy detection of unknown signals over fading channels. IEEE Trans. Commun. 2007, 55, 21–24.

[16] KUMAR, Dilip, ASERI, Trilok C., et PATEL, R. B. EECDA: energy efficient clustering and data aggregation protocol for heterogeneous wireless sensor networks. International Journal of Computers Communications & Control, 2011, vol. 6, no 1, p. 113-124.

[17] T. Wiegand, and G. J. Sullivan, “The H.264/AVC Video Coding Standard," IEEE Signal Proc. Mag., pp. 148-153 Mar 2007.

[18] SHAH, G., ALAGOZ, Fatih, FADEL, Etimad, et al. A Spectrum-Aware Clustering for Efficient Multimedia Routing in Cognitive Radio Sensor Networks. 2014.

[19] BOYINBODE, Olutayo, LE, Hanh, et TAKIZAWA, Makoto. A survey on clustering algorithms for wireless sensor networks. International Journal of Space-Based and Situated Computing, 2011, vol. 1, no 2, p. 130-136.

[20] Wagstaff K, Cardie C, Rogers S, et al. Constrained k-means clustering with background knowledge. In: Proceedings of the 18th International Conference on Machine Learning, 2001 June 28–30, Williamstown. San Francisco: Morgan Kaufmann Publishers, 2001. 577–584

[21] Klein D, Kamvar S D, Manning C D. From instance-level constraints to space-level constraints: Making the mostof prior. In: Proceedings of the 19th International Conference on Machine Learning, 2002 July 8–12, Sydney. San Francisco: Morgan Kaufmann Publishers, 2002. 307–314

[22] SALEEM, Yasir et REHMANI, Mubashir Husain. Primary radio user activity models for cognitive radio networks: A survey. Journal of Network and Computer Applications, 2014, vol. 43, p. 1-16.

[23] H. Kim and K. G. Shin, “Efficient Discovery of Spectrum Opportunities with MAC-Layer Sensing in Cognitive Radio Networks," IEEE Trans. On Mobile Comp., vol. 7, no. 5, pp. 533-545, 2008.

[24] http://cloudns2.wordpress.com/5-1-cognitive-radio-crcn-patch/

[25] http://download.mobnets.rwth-aachen.de/

[26] STELLWAGEN, Eric et TASHMAN, Len. ARIMA: The Models of Box and Jenkins. Foresight: The International Journal of Applied Forecasting, 2013, no 30, p. 28-33.

[27] BRADAI, Abbas, AHMED, Toufik, et BENSLIMANE, Abderrahim. ViCoV: Efficient video streaming for cognitive radio VANET. Vehicular Communications, 2014.

[28] Minitab Inc, 2010. Minitab Statistical Software. Minitab Release 15. http://www.minitab.com.

[29] PEI, Guangyu et CHIEN, Charles. Low power TDMA in large wireless sensor networks. In : Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE. IEEE, 2001. p. 347-351.

[30] CESANA, Matteo, CUOMO, Francesca, et EKICI, Eylem. Routing in cognitive radio networks: Challenges and solutions. Ad Hoc Networks, 2011, vol. 9, no 3, p. 228-248.

[31] The Network Simulator (NS-2), http://www.isi.edu/nsam/ns

[32] http://stuweb.ee.mtu.edu/~ljiali

[33] http://www2.tkn.tu-berlin.de/research/evalvid/EvalVid/docevalvid.html

[34] Priestley, M. B. 1981 Spectral analysis and time series / M.B. Priestley Academic Press, London; New York."OMNeTT++ Discreet Event Simulation System". Internet, 2010 (http://www.omnetpp.org/)