IAC-16,B2,6,x35399

Simulation and Implementation of Cognitive Radio Algorithms for Satellite Communications

Pedro Rodrigues[a]*, Pedro Sinogas[a], Sérgio Cunha[a], Stefan Taing[b], Jens Elsner[b], Mathias Uhlenbrock[b], Pedro Silva[c], Luís Pessoa[c], Mário Ferreira[c], João Canas Ferreira[c], Simon Watts[d]

[a] *Tekever, Portugal, pedro.rodrigues@tekever.com*
[b] *Munich Innovation Labs, Germany, je@munich-innovation.com*
[c] *INESC TEC and Faculty of Engineering, University of Porto, Portugal, pmms@inescporto.pt*
[d] *Avanti Communications Group plc, United Kingdom, simon.watts@avantiplc.com*
* Corresponding Author

Abstract

The massification of mobile access and services has increased the demand for faster, reliable and ubiquitous networks, which has been leading to additional pressure on satellite service providers to provide larger throughput. This inherently raises the challenge of bandwidth management. Regulatory activities have led to frequency allocation charts that are growing more complex and harder to manage. Such problem needs therefore to be addressed in order to achieve a more efficient use of resources and cope with the escalating traffic in satellite communications in the sub 5GHz bands. The H2020 SCREEN project is addressing this challenge by resorting to cognitive radio (CR) technology at S band. SCREEN is working towards maturing several CR enabling technologies up to TRL4/5, considering two reference scenarios: Satcom-enabled UAV constellations and Inter-Satellite Links for satellite networks.

This paper focuses on the design, development, simulation and implementation of the proposed cognitive radio algorithms in SCREEN, namely spectrum sensing, dynamic spectrum manager (DSM), and learning techniques, presenting the most promising results achieved thus far.

In CR environments, communication conditions may show a considerable variability, and therefore, adaptable and reconfigurable spectrum sensing architectures can bring valuable benefits. In this paper, we describe a multi-resolution spectrum sensing architecture, compatible with the proposed approach for dynamic spectrum management, which considers a local and a global DSM and how to combine both methods to offer a higher level of performance. Regarding learning techniques, SCREEN defined two principal strategies: Centralized learning and de-centralized learning that lend themselves to different protocol architectures, namely in terms of medium access control (MAC).

Additionally, a novel simulation framework for evaluating cognitive radio for Satcom applications is presented, which is based on the open source network simulator (ns-3). The simulator considers realistic satellite orbits, propagation loss and propagation delay models and supports the placement of interferer nodes. The simulation output is then tailored for visualization in Google Earth. Such an integrated simulation tool is one of the major novelties of SCREEN. Simulation results will be presented on the comparison of both centralized and decentralized MAC approaches.

Finally, early implementation results of these algorithms in an off-the-shelf space Software-Defined Radio platform will be discussed, as a pioneer step into showing the true applicability of cognitive radio for a new generation of flexible and versatile space-bound transceivers.

Keywords: Cognitive Radio, Satellite Communications, Spectrum Sensing, Simulation

1. Introduction

The SCREEN project has been developing a cognitive radio solution which will enable the efficient use of spectrum resources required to cope with the escalating traffic in satellite communications in the sub 5GHz bands. Two reference scenarios are under consideration: (1) Satcom-enabled UAV constellations and (2) Inter-Satellite Links for satellite networks. Several CR algorithms have been developed, including adaptable and reconfigurable spectrum sensing algorithms, a dynamic spectrum manager consisting of a hybrid local and global approach, and learning techniques consisting of a combination of both centralized and de-centralized learning. The performance of the proposed algorithms and approaches is evaluated in an open source network simulator which takes into account realistic satellite orbits, propagation channel and interferer nodes, and can be visualized in a Google Earth based application.

The paper is structured in 6 sections. Section 2 describes the multi-resolution spectrum sensing architecture employed, section 3 addresses the strategies defined regarding learning techniques, section 4 describes the novel simulation framework and section 5 presents early implementation results of these algorithms. The last section concludes the paper.

2. Spectrum sensing architecture

2.1 Dynamic Spectrum Manager (DSM)

The Dynamic Spectrum Manager (DSM) will have the responsibility to merge and analyse the information coming from the spectrum sensing and learning algorithms, cross-check it with the regulatory constraints and make a reasoned decision about the configuration modifications in the radio. It is the core module of the cognitive cycle handling the crucial and fundamental decision-making process. Considering the techniques which can be used on a SDR platform, based on Consortium experience, the SCREEN approach to address this functionality is split in two compatible supervisors: Local and Global DSM. The first one aims to operate within the transceiver bandwidth at a specific central frequency, so it has limited range of operation, but doesn't require a full radio reconfiguration. On the other hand, the global DSM aims to operate within the entire transceiver range of operation, by changing the central frequency to a different slot within its radio reconfiguration capability.

The division of this module in two methods increase the accuracy and speed up the decision-making process and takes advantages of the SDR capability and of the FHSS techniques currently used in the GAMALINK platform selected for the implementation (see Section 5), as follows:

- **Local DSM**: takes advantage of the FHSS to overcome partial and temporary degradation of the channel conditions throughout its bandwidth.
- **Global DSM**: takes advantage of the SDR capability and changes the central frequency to a different slot within its reconfiguration capability.

Although each method can be executed separately and alone, these two methods will be combined to offer a higher level of performance, while avoiding permanent full reconfigurations of the SDR. The local DSM is based on sensing results of the spectrum within a limited frequency range, normally within the transceiver bandwidth, around the nominal frequency of operation of the transceiver and auxiliary by the learning insights. Local DSM manages non-permanent or partial degradation of the channel in the receiver bandwidth without the need to reconfigure the radio and, through a decision-making process, it decides which frequencies, within the transceiver bandwidth, will be used for the frequency-hopping.

However, in some cases, the entire band where the transceiver is operating may be subject to interference and the local DSM cannot solve the problem. In such cases, the global DSM can reconfigure the SDR platform to operate in a different central frequency, chosen based on wide-spectrum sensing results and supported by learning method outcomes. In the new operating frequency, local DSM can operate again, over the transceiver bandwidth but now centralized at the new frequency.

2.2 Wideband spectrum sensing techniques

The next generation of wireless communications aims to significantly improve data throughputs, system capacity and spectrum usage. This obviously applies to Cognitive Radio systems and will require the detection and exploration of more unused spectrum bands within wider frequency ranges, in order to achieve higher aggregated throughputs. Under these circumstances, traditional narrowband spectrum sensing techniques may not achieve the desired performance. So, wideband spectrum sensing algorithms, capable of detecting white spaces over frequency bands ranging from hundreds of MHz to several GHz, have been proposed.

Algorithms as multiband joint detection [1] and wavelet-based spectrum sensing [2] were proposed, based on wideband signal sampling with an ADC followed by digital signal processing procedures (such as FFT). In multiband joint detection a wideband signal is treated as a set of narrowband signals processed independently (Fig. 1).

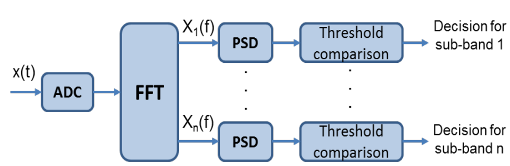

Fig. 1 - General architecture of the multiband joint detection algorithm

In turn, wavelet detection performs spectrum holes detection based on discontinuities observed on the wideband PSD wavelet (Fig. 2).

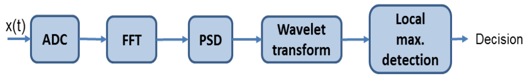

Fig. 2 - General architecture of the wavelet detection algorithm

Although conceptually simple, these methods require the sampling rate to be at least twice the maximum frequency of the signal to be sampled - *Nyquist frequency*. Consequently, in high-speed signal scenarios, ADCs with considerable high data rates, resolution and thus, power consumption will be needed.

A solution to overcome the implementation challenges imposed by high frequency ADCs consists on the use of *Hybrid Filter Banks* – HFB [3], [4]. A Hybrid Filter Bank comprises an *analysis filter bank* - analogue domain - and a *synthesis filter bank* - digital domain. In this kind of systems, the input signal is frequency multiplexed into several bandpass channels. Then, each channel is sub-sampled and converted to the digital domain.

Sub-Nyquist techniques -in which the acquisition of the incoming RF signal is done using sampling rates - have been proposed and investigated. Due to inefficient utilization, the spectrum presents a considerable degree of sparsity and information about its usage profile can be compressed. Based on this observation, *compressive sensing* was proposed by Tian and Giannakis[5]. In their work, sampling rates lower than the Nyquist frequency - *sub-Nyquist* rates - were employed to obtain coarse-grain information about spectrum usage. After signal reconstruction, spectral opportunities are identified through wavelet-based edge detection. In contrast with earlier works on compressive sensing, Tropp et al [6] extended this technique to analogue signals by proposing the *Analog-to-Information Converter* (AIC) approach. One of the AIC drawbacks is its high sensitivity to design imperfections or model mismatches. In order to mitigate this problem, Mishali and Eldar [7] modified AIC into a parallel channel model - Modulated Wideband Converter. Another Sub-Nyquist sensing technique proposed is Multi-rate Sub-Nyquist sampling [8].

Most Sub-Nyquist compressive sensing techniques were designed under the assumption that the spectrum usage is sparse. But, as Cognitive Radio networks evolve, spectrum utilization may considerably improve and the previous assumption may not hold any more. This issue is mentioned in [9] as a challenge which will demand sub-Nyquist sensing techniques suitable for non-sparse spectrum usage. Other research challenge for sub-Nyquist sensing identified in [9] has to do with the ability to adapt to time-variant sparsity levels - Adaptive sensing.

More generally, the reliability of spectrum sensing algorithms could be improved by exploiting spatial diversity of spectrum usage profiles across different CR devices - *Cooperative Wideband* [10].

3. Learning techniques

In this chapter the learning techniques used for adaptive transfer channel attribution are described. After thorough consultation of the relevant literature, Q-learning has been identified as a possible candidate for a cognitive algorithm approach. The chapter is organized as follows: in section 3.1 the Q-learning algorithm is described in some detail and in section 3.2 the individual steps which are planned in order to validate the algorithm and evaluate its performance are outlined.

3.1 The Q-learning algorithm

Q-learning belongs to the larger class of unsupervised learning methods and therefore a dedicated training of the algorithm prior to its application is not required. Furthermore, it is a reinforcement learning technique, which basically means that desirable algorithm behaviour is supported by providing some kind of reward, whereas objectionable behaviour is sanctioned. By these means, an optimal action strategy can be learned based on maximization of the total reward.

More specifically, the algorithm defines a set of states S and a set of actions A which can be performed to move between states. The execution of an action which results in a specific state is evaluated and the degree of success is measured by a numerical score which is also called the reward. The goal of the algorithm is now to maximize its total reward. Ideally, this is achieved by learning which action is optimal for each state. Hereby, the action that is optimal for each state is the action that promises to score the highest so-called long-term reward. This reward is a weighted sum of the expected values of the rewards of all possible subsequent steps starting from the current state. The weight for a step N steps into the future is calculated as γ^N ($0 \leq \gamma \leq 1$), where γ is called the discount factor and trades off the importance of sooner versus later rewards.

The core of the algorithm is a function Q that calculates the quality of a state-action combination:

$$Q: S \times A \rightarrow \mathbb{R}$$

which maps a state-action pair (s, a) onto the set of real numbers $\mathbb{R}$. Initially, Q returns an arbitrarily chosen value. Then, each time an action is selected by the algorithm a reward is provided and Q is updated. With other words, the algorithm consists of a simple value iteration update. The new quality factor Q is then calculated as

$$Q(s_t, a_t) = Q_{prev} + \alpha_t \cdot [r_{t+1} + \gamma \cdot \max_a Q(s_t, a) - Q_{prev}]$$

where r_{t+1} is the reward, provided after the action a_t is performed at time t and being in state s_t and where $\alpha_t = \alpha(a_t,\ s_t)$ $(0 \leq \alpha_t \leq 1)$ is the learning rate. The learning rate determines to what extent the newly acquired information will override the old information, with $\alpha = 0$ representing no learning and $\alpha = 1$ representing the situation where only the information from the immediately preceding step is taken into account, resulting in maximally fast adaption. Q_{prev} is the function value at the previous iteration $Q_{prev} = Q(s_{t-1}, a_{t-1})$ and γ is the discount factor as described in the preceding section. If the desired behaviour consists of a finite number of steps, at some point, the algorithm will reach a so-called final state s_f. For all final states s_f $Q(s_f, a)$ is never updated but is set to the reward value r. In most cases, $Q(s_f, a)$ can be taken to be equal to zero.

For the static case of satellites moving (infinitely) within fixed orbits, no such final state exists. However, if after a certain time span T, the overall situation is exactly the same, or, with other words, the environmental conditions change periodically, the problem can be divided into so-called episodes. To some extent, the last state of an episode plays then the same role as a final state in the finite problem. Here, the state is modelled as a tuple of values, representing e.g. the actual channel number and a measure which depends on the number of successful and failed packet transmissions during the last n iterations.

If, on the other hand, the environmental conditions change unpredictably, as it is probably the case in any real world scenario, the Q-learning method can still be used as an adaptive algorithm. In contrast to the situation outlined above the values of the Q-Matrix will never converge. Given a sufficiently high learning rate, however, the algorithm is still able to adequately react depending on the evaluation of the current state in order to choose a strategy which is expected to provide better results than a less optimal approach such as e.g. random channel attribution.

3.2 Algorithm validation and performance evaluation

The individual steps in order to validate and evaluate the cognitive algorithm approach are planned to be executed in the following order: After implementation, the algorithm is tested on a very simple but not trivial scenario comprising only a small number of satellites and ground stations. However, even in such a simple scenario, at least one communicating node should be forced to use a channel with considerable interference at some point in time. In this case, the algorithm should be able to learn when the transition to such a noisy channel represents the optimal choice in the given situation. Consequently, the algorithm is deemed to perform successfully, if e.g. the number of dropped packets due to interference is smaller than e.g. by random channel attribution. Concomitantly, the performance of the algorithm is planned to be compared to a simple channel hopping heuristic which is expected to greatly help with the interpretation of the results. Departing from there, more satellites and ground stations can be added, with the aim of creating scenarios which exhibit much more involved interference patterns. Depending on the observed learning rate, scenarios with non-periodic perturbations, which e.g. can be represented by UAVs which are active only for a limited period of time, can be investigated in addition.

4. SCREEN Simulation Framework

This section describes a novel simulation framework based on the ns-3 discrete-event network simulator [11], and developed to evaluate cognitive radio for Satcom applications. It considers realistic satellite orbit prediction, propagation loss and propagation delay models, and supports the placement of LEO and GEO satellites, ground stations, and interferer nodes (e.g., UAVs). In addition, the framework comes with scripts to convert the raw simulation output into a format which enables graphic visualization within Google Earth.

The remainder of this section is structured as follows. Section 4.1 provides an overview of the simulation framework. Section 4.2 discusses the interface used to evaluate cognitive algorithms, and how new cognitive algorithms can be implemented. Section 4.3 presents the satellite module used to predict satellite orbits. Section 4.4 describes the supported node types, how their behavior is modelled, and the supported mobility models. The physical layer and wireless channel abstractions are presented in Section 4.5. The visualization of the simulation results is depicted in Section 4.6.

4.1 Overview

The SCREEN simulation framework is based on the ns-3 network simulator, and thus it is written in C++. It implements packet-based communications between satellites and ground stations to enable cognitive radio evaluation, and supports the placement of interferers. Cognitive algorithms provide the channel and the modulation scheme to be used when offloading data packets, and new algorithms can be added by implementing the provided interface. Cognitive algorithms are described in more detail in Section 4.2.

The framework is designed to enable cognitive algorithm evaluation without writing a single line of code, if using an existing cognitive algorithm. Thus, the scenario configuration and the Earth orientation parameter (EOP) values required to accurately predict satellite orbits (Section 4.3) are defined through input text files, enabling seamless configuration and update. The simulation scenario is configured by defining a set of global simulation parameters, such as the cognitive algorithm to be used and the set of channels available, and by specifying the required simulation nodes and their individual properties, such as antenna gains and mobility model. The simulation results are written to an output file for further visualization.

We assume the following. A satellite chooses one of the visible ground stations at random, if any, from the set of configured ground stations. There is a dedicated channel pair ($P_{downlink}/P_{uplink}$) available to initiate communications between a satellite and a ground station. A packet is considered to have been successfully received if the SINR is always above threshold during transmission, which depends on the modulation scheme being used and on the coding ratio. There are four supported modulation schemes – BPSK, QPSK, 16-QAM, and 64-QAM –, and the coding ratio is ½ by default.

The protocol used to offload data packets from satellites to ground stations works as follows. A satellite sends an offload request over $P_{downlink}$ to the ground station with the number of packets to offload. If the ground station is idle and the request is successfully received, it sends a reply over P_{uplink} with the channel and modulation scheme to be used when transmitting data packets. The network cards of both nodes are then switched to the selected channel, and the data packets are offloaded. If the ground station is either sensing or already handling an offload request, the incoming offload request is ignored. The node properties and behaviour are described in more detail in Section 4.4.

The wireless module encompasses the functions developed to abstract the wireless channel and the physical layer. It enables to configure existing ns-3 propagation loss and propagation delay models, which are realistic and widely used. The wireless module considers packet transmission overlapping, in which case the reception is attempted for the packet with higher SINR. This module is presented in more detail in Section 4.5.

The simulation output provides periodic sampling of noise and interference values across all available channels, and information about incoming packets: source, destination, their geolocations, the SINR, the modulation scheme, and the reason for dropping a given packet (if it has been dropped). This information can be visualized within Google Earth, where a satellite symbol is drawn along its trajectory for each (every n-th) communication attempt. The details of the communication attempt are summarized within a popup window, which can be accessed by clicking on the respective satellite symbol. More details on the simulator visualization can be found in section 4.6.

4.2 Cognitive Algorithms

The SCREEN simulation framework defines an interface that cognitive algorithms must implement in order to support seamless integration of new cognitive algorithms. Cognitive algorithms determine the channel and modulation scheme to be used, and estimate the noise and interference of channels through the SINR of incoming packets and through the schedule of sensing events. The interface defines four mandatory functions: 1) a function that returns the best channel and the modulation scheme to be used; 2) a function to schedule sensing events that expects the channel, time duration, and the amount of samples to be collected as parameters; 3) a callback function that provides the SINR of incoming packets immediately after being received or discarded; 4) a callback function to return the sensing results.
For example, in the current implementation of the Q-learning algorithm, the numerical score which is used to measure the immediate reward resulting from being in state s is inversely proportional to the noise and interference found in this state. This means, that the higher the level of noise and interference which is found in a certain channel, the lower the reward which will be attributed to the corresponding state.

4.3 Satellite Module

The satellite module encloses all the required functions to accurately predict satellite orbits. The satellite mobility model is configured using the Two-Line Element (TLE) format, developed by NORAD (North American Aerospace Defense Command) as a compact means to obtain modestly accurate and fast calculations [12] due to the impossibility of continuously track each single object in space. It is based on the SGP4/SDP4 program codes [13] because the largest database available for artificial objects orbiting the Earth is generated using these models, (2) the satellite orbit prediction using the TLE data format has maximum accuracy using these models, (3) the program codes are in actual use within DoD (U.S. Department of Defense) and revised versions have been released over the years, and (4) the orbit calculations are fast while the error is approximately 1 km, as long as the proper TLE data file(s) is/are used. The output of SGP4/SDP4 is in *true equator, mean equinox* (TEME) coordinate frame, an Earth-centered inertial (ECI) frame, and therefore it needs to be converted into an Earth-centered Earth-fixed (ECEF) frame.

Astronomers typically measure time in Julian days, the number of solar days since the beginning of the Julian Period (12h, January 1, 4713 BCE), and several astronomic formulas expect time to be measured in such way or using derived epochs, such as Modified Julian Date (MJD) – days since 0h, November 17, 1858 –, and J2000.0 – days since 12h, January 1, 2000. For this reason, several SGP4/SDP4 functions expect time to be expressed either in Julian days or using the TLE epoch. We follow the same approach as Vallado et al. in [13], and consider time in Julian days to be realized as UTC except for Greenwich Mean Sidereal Time (GMST) calculation where UT1 is considered. Their approach is justified by the fact that it is unknown whether UT1 or UTC is what it is required by the program codes, and that the error associated with approximating UT1 with UTC (DUT1 = UT1 – UTC) is, at most, ±0.9s, which is within the theoretical uncertainty of the SGP4 theory itself [13]. Therefore, time needs to be converted, at least, from UTC to Julian and vice-versa.

This module encloses coordinate frame conversion functions that are capable of converting TEME into *International Terrestrial Reference Frame* (ITRF), an ECEF coordinate frame. This way, not only coordinates can be converted into WGS84 format (latitude, longitude, and altitude) but are also compatible with ns-3 internal coordinate system. Still, to convert TEME to ITRF, several EOP parameters are required and whose values may have an impact on the accuracy of the SGP4/SDP4 models. Therefore, when compiling the SCREEN simulation framework, the EOP parameters are read from two files that are made weekly available by the U.S. Naval Observatory (USNO). These files provide a compilation of EOP parameters that includes past measurements (since 1 January 1992) and predictions for several months ahead. The timescale conversion is also built upon those files to accurately convert between the following timescales: Julian, UTC, UT1, TAI (International Atomic Time), TT (Terrestrial Time), GPST (Global Positioning System Time), and Unix/POSIX time systems. Internally, time is kept using Unix epoch and can be defined up to the millisecond, which is also the precision of TLE data.

The satellite mobility model is configured using the TLE data format, and the simulation start timestamp. ns-3 network simulations typically use solely time relative to simulation start, and therefore the simulation start timestamp enables to convert relative time into absolute time. When needed, absolute time is realized as UTC.

4.4 Simulation Nodes

The SCREEN simulation framework supports four types of nodes: LEO satellite, GEO satellite, ground station, and interferer (e.g., a UAV). The behaviour of satellites and interferers is modelled using a two-state Markov model that defines the state transition probabilities, and the periodicity of such transitions. For the satellite nodes, the number of packets to be offloaded follows a Poisson distribution. The state transition probabilities, their periodicity, and average number of packets for the Poisson distribution are set on the scenario configuration file.

We consider four mobility models that enable to configure fixed geolocations, satellite orbital positions (LEO and GEO), and waypoint paths that can be set in loop. The static mobility model is defined through fixed latitude, longitude, and altitude values, and is mostly used for ground stations. The LEO mobility model is defined through the TLE format, and is used for LEO satellites. The GEO mobility model is akin to the static mobility model to avoid unnecessary calculations because geostationary satellites are only able to maintain their position by performing periodic corrections (*stationkeeping*), but it is defined through the TLE format. The waypoint mobility model is mostly used for interferers, and enables to set a path of points along the time elapsed to go from one point to the next one. This mobility model supports closed loops.

The antenna properties – noise factor, Rx and Tx gains –, and the Tx power can be set on the scenario configuration file.

4.5 Wireless Module

The abstraction of the physical layer and the wireless channel provided follows the same approach of the module used to simulate 802.11 on ns-3. Whenever a packet is sent, all the nodes within the same channel receive a delayed notification with the packet, its transmission duration, and the Rx power: the delay is evaluated using the propagation delay model, and the Rx power is determined using the Tx power and the propagation loss model; we also consider line-of-sight (LoS) for the node to be reachable. By default, we use a constant delay propagation delay model, and Friis propagation loss model. The incoming packet notifications create a new interference event, unless the node is the intended recipient and the SINR is above threshold. The interference is then calculated considering all interference events that are active.

Packet transmissions may overlap, in which case the reception is attempted for the packet with higher SINR. For incoming packets targeting the node, when a packet is dropped, it is provided a flag that indicates the reason for the packet drop. We consider the following cases: SINR is below threshold; the SINR of the current packet is lower than the SINR of the incoming overlapping packet; SINR of the incoming overlapping packet is lower than the SINR of the current packet. It is created an interference event for the packet being dropped due to transmission overlapping, whose duration is the time left to end the transmission.

We developed and implemented a fast LoS algorithm that approximates the Earth to a sphere. It works as follows. Let A, B, and O be the 3D coordinates of, respectively, node A, node B, and the centre of the Earth. Let also r be the line defined by vector $\overrightarrow{AB}$ normalized. Thus, the point of r in parametric form, r(t), that is closest to O is given by $t = \overrightarrow{AB} \cdot \overrightarrow{AO} / \|\overrightarrow{AB}\|^2$. The distance d between O and that point is given by $d = \|\overrightarrow{AB} \times \overrightarrow{AO}\| / \|\overrightarrow{AB}\|$. There is always LoS unless $0 \leq t \leq 1$ and d is less than the Earth radius. The first condition implies that the closest point to O is between the two nodes, and the second condition implies that the point must be below the surface. We use the polar radius as it is around 31 km shorter than equatorial radius.

4.6 Visualization

In order to provide graphic visualizations of the simulator output, the framework comes with a couple of scripts written in Python which convert the raw text format into KMZ, which can then be opened within Google Earth. The focus of the visualization is on capturing information on the local communication environment which is primarily given by the relative position of the transmitting and receiving nodes. In this manner, each line in the raw simulator output represents a communication attempt between two nodes. In the visualization, the 3D-position of a node is depicted by a symbol which distinguishes between the various types of nodes, e.g. satellites, ground stations or UAVs. In addition, the trajectory of moving nodes, e.g. low earth orbit satellites and UAVs, is also shown. Furthermore, a time stamp is attached at each node position. Within Google Earth this information enables a slider functionality, which can be used to animate node movement. For more information about the current communication attempt a popup window can be toggled by clicking on the respective symbol. Here, the viewer finds a complete summary of all parameters which are being put out by the simulator. These are, in no particular order: identifiers of the transceiver and the receptor, the current time stamp, the modulation of the current packet, the received power of the current packet in Watts, the SINR and the level of noise and interference of the current packet in dBs, both separated according to the current transfer channel and the attributed drop flag.

5. Implementation

5.1 Transceiver selection

The SCREEN project does not intend to design and develop a new SDR platform, but, instead, to use an existing one and adapt it to integrate the cognitive algorithms. Being readily available and also successfully launched into space, the GAMALINK platform is a good candidate to fit within this approach. This multifunctional platform provides great flexibility and modularity to be suitable for several specific missions with minor modifications. Furthermore, it avoids wasting time and resources on designing and developing a new SDR platform and allows speeding up the technology maturity of the cognitive radio concept. Because of that, the selection of GAMALINK as the baseline transceiver to be used for SCREEN seems to be optimal and feasible, when looking at the SCREEN objectives.

The hardware communications platform of GAMALINK is based on Software-Defined Radio (SDR), a technology that enables the development of various waveforms using a common hardware platform. Its characteristics can result in tremendous mass and volume savings, while increasing flexibility to a point where a radio system could be completely modified by just sending a command from ground.

Moreover, it allows the operation of different subsystems simultaneously in the same hardware such as radio communications, Global Navigation Satellite System (GNSS) reception for navigation and position and attitude measurement. The core specifications of the GAMALINK platform are presented below.

Core GAMALINK Specifications	
Dimensions	95.9 x 90.2 x 11 mm (PC104)
Mass	<100 g
Supply voltage	3.3V or 5V
Data interface	I2C, UART
Storage capacity	from 2 x 2 GB
Frequency range	300 MHz to 3 GHz SDR
Networking	Ad hoc networking for constellations

Table 1 – Core GAMALINK's technical specifications.

5.2 Implementation resource estimation

As a first step to define the modifications needed to make the GAMALINK platform suitable for SCREEN, the CR algorithms required resources were estimated to evaluate the capacity of GAMALINK to incorporate these algorithms. Table 2 presents an estimation of approximate minimum performance values concerning memory, required processing, response time and implementation target block, extracted from the algorithm studies.

CR module	Memory	Processing	Response time	Implementation target
Decision Making	-	Look-up / simple search	10 ms	GPP
Spectrum Sensing	Up to 1024 bin FFT on FPGA with averaging: 8 – 16 kB	Thresholding and detection	1 ms - 100 ms according to averaging duration	FPGA / DSP
Learning	1 kB	Depending on algorithms	10 ms	GPP
Reconfiguration	100 kB	-	100 ms	GPP / FPGA
Regulatory /Other Constraints	10 kB	Depending on algorithms	1000 ms	GPP

Table 2 – CR algorithms resources estimation.

Memory requirements for learning and decision making stem from the fact that at least a channel state vector has to be kept. Reconfiguration, if need be, will involve reprogramming of part of the digital section and is associated with higher memory requirements. Regulatory constraints should be accessible in the form of a small database or text file. Processing capabilities cannot be exactly stated, as these heavily depend on the employed algorithms. The response times should correspond to maximum delay round-trip times or be close to the minimum packet lengths employed in the communication protocol. The implementation target designates the likely building block that can carry out the required function optimally.

5.3 Implementation requirements

Regarding the hardware system, it is important to consider that the transceiver to be used within SCREEN project shall have at least the following three capacities:

- Computational power available in the processing core.
- RF reconfiguration capability
- RF reconfiguration agility

In terms of software, the local DSM takes advantage of the already-implemented frequency-hopping technique in GAMALINK to avoid a dedicated sensing mode, but it requires certain capabilities at waveform and protocol level, such as:

1) The sequences of frequency-hopping are required to be dynamically allocated.
2) The receiver needs to perform and store measurements of each individual frequency of the FHSS and sends back such information to the transmitter, so that it is aware of which frequencies are "good" and which are "bad", as well as their utility level.
3) The transmitter needs to decide upon the frequency slots to use and shares it with the receiver, so that it is able to interpret the transmitted messages.

On the other hand, the global DSM only has an impact at operation level:

4) Needs to be integrated into the transceiver local management module.
5) Wide-spectrum sensing requires a dedicated operating mode.

5.4 GAMALINK modifications

In terms of hardware and from the estimated required resources in the previous sections, it was concluded that all processing core components in GAMALINK can implement the CR algorithms. Thus no more computational power is required for the transceiver. GAMALINK already proved its capability and agility to reconfigure the radio due to its baseline SDR technology. Such capability shall be sufficient to handle the Decision Making approach, in both local and global DSMs. Finally, with respect to RF reconfiguration agility, GAMALINK can move between frequencies with a latency in the order of micro-seconds and reconfigure the entire SDR in the order of mili-seconds. Therefore, we conclude that no hardware modifications to GAMALINK are needed for the testing purposes in SCREEN.

In terms of software, starting from the global DSM needs, it is easy to conclude that no deep modifications are required. As it was decided on CR module distribution, the Decision Making algorithms will be implemented on the main operation function, which is responsible for performing the transceiver management. Furthermore, and taking advantages of its flexibility, to add a dedicated operating mode is a task easy to handle.

Concerning the local DSM requirements some lower level modifications are required that may imply a large number of resources allocated and high levels of effort committed. The two last requirements are crucial for a correct operation of the communication link between two nodes of a network. The main required modification concerns the RF protocol, which has to incorporate the information that needs to be exchanged between the two nodes. Therefore, the RF protocol has been modified to include not only the sensing results from a receiver but also the decision making outcomes from the transmitter.

The last but not least important requirement relates to the dynamic FHSS allocation, which is required to optimise the channel quality. Currently, GAMALINK provides static sequences of frequency hopping within its bandwidth and therefore without a deep modification of the waveform, it cannot comply with this requirement. For this reason, it has been decided that the requirements 2) to 5) will be implemented for the SCREEN validation campaign, but requirement 1) will not be implemented. This however does not have an impact on the validation of the algorithms, only a loss of performance.

6. Conclusions

This paper presented the key results of the design and implementation stages of cognitive radio algorithms, studied in the frame of the H2020 SCREEN project. Implementation is expected to be completed very shortly, in order to perform the testing activities before the end of the year. In terms of testing, some work is already being carried out to obtain satellite data and noise models to emulate Satcom links, to better evaluate the algorithms performance in its operating environment.

The development of the SCREEN simulator, which integrates several simulation tools, such as propagation models, orbit dynamics and cognitive techniques, is seen as an important tool to be exploited after the end of the project alongside with the actual algorithm implementations.

Preliminary simulation and implementation results have pointed out the potential of cognitive techniques in radio communications and the overall benefits that it can bring to the future generation of Space constellations. They will bring major challenges in terms of interference management or spectrum regulations. The work carried out in SCREEN is maturing state of the art technologies to bring new tools to the table, for mission designers to use in the next generation of Space exploitation.

Acknowledgements

The work presented herein has been funded by the European Community Horizon 2020 Space Programme (H2020-SPACE) under Grant Agreement 640210. The SCREEN project (Space Cognitive Radio for Electromagnetic Environment maNagement) is a project funded under the topic "COMPET-6-2014: Bottom-up space technologies at low Technological Readiness Level", involving 4 partners and started on January 1st 2015.

References

[1] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, 'Optimal Multiband Joint Detection for Spectrum Sensing in Cognitive Radio Networks', *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1128–1140, Mar. 2009.

[2] Z. Tian and G. B. Giannakis, 'A Wavelet Approach to Wideband Spectrum Sensing for Cognitive Radios', in *1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2006.*, 2006, pp. 1–5.

[3] D. F. Albuquerque, J. M. N. Vieira, N. B. Carvalho, and J. R. Pereira, 'Analog Filter Bank for Cochlear Radio', in *2010 IEEE International Microwave Workshop Series on RF Front-ends for Software Defined and Cognitive Radio Solutions (IMWS)*, 2010, pp. 1–4.

[4] J. P. Magalhaes, J. M. N. Vieira, R. Gomez-Garcia, and N. B. Carvalho, 'Bio-Inspired Hybrid Filter Bank for Software-Defined Radio Receivers', *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 4, pp. 1455–1466, Apr. 2013.

[5] Z. Tian and G. B. Giannakis, 'Compressed Sensing for Wideband Cognitive Radios', in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007.*, 2007, vol. 4, p. IV-1357-IV-1360.

[6] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, 'Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals', *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, Jan. 2010.

[7] M. Mishali and Y. C. Eldar, 'Blind Multiband Signal Reconstruction: Compressed Sensing for Analog Signals', *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 993–1009, Mar. 2009.

[8] H. Sun, W.-Y. Chiu, J. Jiang, A. Nallanathan, and H. V. Poor, 'Wideband Spectrum Sensing With Sub-Nyquist Sampling in Cognitive Radios', *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 6068–6073, Nov. 2012.

[9] H. Sun, A. Nallanathan, C.-X. Wang, and Y. Chen, 'Wideband spectrum sensing for cognitive radio networks: a survey', *IEEE Wireless Communications*, vol. 20, no. 2, pp. 74–81, Apr. 2013.

[10] G. Ganesan and Y. Li, 'Cooperative spectrum sensing in cognitive radio networks', in *2005 First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, 2005, pp. 137–143.

[11] ns-3, 'ns-3'. [Online]. Available: https://www.nsnam.org/. [Accessed: 06-Sep-2016].

[12] D. Vallado and P. Crawford, 'SGP4 Orbit Determination', in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, American Institute of Aeronautics and Astronautics.

[13] D. Vallado, P. Crawford, R. Hujsak, and T. S. Kelso, 'Revisiting Spacetrack Report #3', in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, American Institute of Aeronautics and Astronautics.