T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Information Technologies

DEVELOPING LOW ENERGY CONSUMPTION SOFTWARE DEFINED RADIO USING INTERNET OF THINGS

Liqaa Mohammed Hasan

Master of Science

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2021

DEVELOPING LOW ENERGY CONSUMPTION SOFTWARE DEFINED RADIO USING INTERNET OF THINGS

by

Liqaa Mohammed Hasan

Information Technologies

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “DEVELOPING LOW ENERGY CONSUMPTION SOFTWARE DEFINED RADIO USING INTERNET OF THINGS” prepared and presented by Liqaa HASAN was accepted as a Master of Science Thesis in Information Technologies.

____

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of Engineering and Natural Sciences, Altinbas University	___________________
Asst. Prof. Dr. Ahad ARDABILI	School of Engineering and Natural Sciences, Altinbas University	___________________
Asst. Prof. Dr. Sewale Musadaq TAHA	School of Computer Programming Department Beykent University	___________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Liqaa Mohammed Hasan

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through whole research work as well as my family for always assisting me in my hard time.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better realize the technical need for this research work. He helped me by looking outside of all the machine learning and network-based information, and considering the cryptocurrency-based knowledge from a more holistic perspective than just from the productivity point of view. He patiently explained to me a great number of technical details with regards to block-chain and cryptocurrency, for example how to deal with solving problems and to provide insight into the reliability of those results. Without his time and help none of this research would have been possible, and I’m grateful to him for giving me the opportunity to work on this interesting subject. I am also very thankful to my parents as well as my friends who always supported me through whole of this journey.

ABSTRACT

DEVELOPING LOW ENERGY CONSUMPTION SOFTWARE DEFINED RADIO USING INTERNET OF THINGS

HASAN, Liqaa,

M.Sc., Information Technologies, Altinbas University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 05/2021

Pages: 57

In the present research, we focused on developing low energy consumption software defined radio using internet of things so simulated using the MATLAB Simulator is used as a substitute for a real-world cognitive engine. While throughput is often the optimization objective of interest, latency is the performance metric under consideration here. Within a cognitive power consumption swarm, low-latency is likely to be a critical requirement in intra-swarm communication. In order for the platforms to accomplish the mission, messages sent between them will likely need to be delivered with an upper bound on their delay. For example, if one of the platforms in the swarm is to strike a target, rules of engagement may require the firing platform confirm its location with another swarm member before employing ordinance. If no guarantee can be given on the time frame in which the first platform may expect a response, the opportunity to strike the designated target may be missed. Conversely, it may be that the possibility of the SDR returning a solution that leads to poor performance is acceptable where the maximum voltage recorded at 77.18 volts and thus this probability must be minimized in 4900 seconds. An SDR network that implements IoT based genetic algorithm from which actual latency information may be obtain will enable this

work. These experiments should focus on determining whether or not the high probability of the engine finding a highly fit solution as with the experiments presented here carries over to high noise floors with final power output at 0.25W. If not, increased interference may the engine to begin discerning between truly fit solutions and those that would likely not lead to acceptable performance in real-world networks in which the voltage reduction measured at 17.69 volts during experiment.

Keywords: Power consumption; optimization; IoT; software-defined radio; performance transmission; genetic algorithm

LIST OF CONTAINS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

ADC	:	Analog to Digital Converter
CR	:	Cognitive Radio
DOD	:	Department Of Defense
CR	:	Cognitive Radio
DSP		Digital Signal Processing
DSA	:	Dynamic Spectrum Allocation
DAC	:	Digital to Analog Converter
GHz	:	Giga Hertz
PM	:	Project Management
RF	:	Radio Frequency
QoS	:	Software Defined Radio
SCA		Software Communication Architecture
VSWR		Voltage Standing Wave Ratio
USRP		Universal Software Radio Peripheral

1 INTRODUCTION

The concepts of power consumption in software defined radio using internet of things based modulation scheme and order, bandwidth and signal power (referred to hereafter as simply "power") are important to the discussion that follows and thus each warrants a brief introduction here. Power is perhaps the easiest of the three to understand and corresponds to the amount of energy with which a signal is transmitted. The power with which a signal is transmitted affects how far it may propagate and still be correctly received. As the signal propagates through the atmosphere, it is attenuated (loses energy) by interactions with molecules in the air. Thus, as the transmitter and receiver move farther apart, all other things being equal, the signal must be transmitted with more power in order to be received.

SDR enables a flexible approach to support a wide range of wireless communication standards by function designs and configurations. However, lots of expert experience and architecture art are required for mapping and scheduling functional tasks onto the embedded target platform. Moreover, during searching about an optimized design from a series of options, large amount of quantitative analysis about the demand of each task and available resource on the platform are involved in evaluation of the advantages, disadvantages, potential issues and problems for each option as mentioned in [1]. Data-flow modeling is a widely used model-based analysis approach for signal processing applications. Many tools can provide the simulation based on the SDR model to improve the quality and efficiency of the analysis. With the simulator, engineers are able to evaluate the design options, critical problems can be detected before the implementation phase. Therefore, the implementation of the SDR receiver in this project will benefit from data-flow modeling and simulation. The approach is explored to improve the quality and efficiency of the design procedure in industry.

The demodulation and decoding subsystem in the SDR receiver are defined by the standard. A signal processing simulation chain was designed and tested according to the SDR standard [2], with several issues investigated and solutions explored in MATLAB. Another subsystem for demultiplexing is defined in the standard.

In the research about architecture design for the radio receiver on an SDR platform, a flexible architecture is proposed in to support multiple channels in the FM band and two single tuners for DAB, which is a pure digital radio standard in Turkey. To deal with the complexity in the design procedure, many researches have explored data-flow based analysis and design. A reconfigurable phase-shift keying system as an SDR application was designed and implemented with a light weight data-flow approach [3], which facilitates the processes of cross-language, cross-platform migration and prototyping in signal processing application domains for efficient implementation.

The Synchronous Data-Flow (SDF) is an important type of the data-flow in application modeling. But it is hard to describe the dynamic behavior of the SDR system, since the SDF is a restriction of Kahn process networks [4], where actors produce and consume a fixed number of data items per firing with static scheduling. In order to address the limitation in Data-flow based modeling, the dynamic data processing behavior is divided into a group of static modes of operation, where each group is modeled by a SDF graph referred to as a scenario [5], and the worst-case throughput analysis is applied for the scenarios. However, the dependencies between scenarios are lack in the scenario-based data flow model, it will lead to an invalid worst-case temporal analysis. Then a technique for modeling the dependency with Finite-State machine is explored, and the results of throughput were compared and validated with that by the previous scenario-aware data-flow model.

As an extension of the SDF modeling, the Cyclo-Static Data-Flow (CSDF) is much more expressive, it supports algorithms with a cyclically changing, but predefined behavior [6]. The data-dependent and state-dependent conditional behaviors can be described by the CSDF. A static scheduling can be done with CSDF, because the behavior changes with a group of predictable sequences.

Because the SDR standard supports multi-transmission modes and spectrum modes, the CSDF is suitable to describe the signal processing tasks which fires in a periodical and data-dependent pattern. In this report, the SDR receiver will be modeled by the CSDF, the architecture options will be applied on the models and simulated with a data-flow simulator to analyze and generate optimal design options.

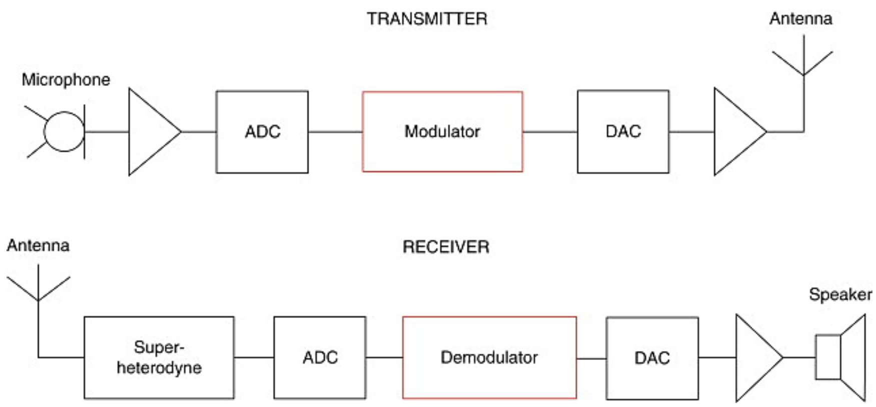


Figure 1.1: Depicts the transmitter and receiver structure in the SDR [5].

An SDR system diagram includes both transmitting and receiving systems. For the SDR transmitter one type of the input to the transmitter is radio program containing multiple audio streams and data services, which is defined as the Main Service Data (MSD) in the SDR standard. The other type of input is Service Description Information (SDI) defined to carry the program guide information, configuration of the MSD and network information [7]. The multiplexing subsystem generates Service Multiplexed Frames (SMF) with MSD bits and Control Multiplexed Frames (CMF) with the SDI bits. System Information (SI) is required to figure out the configuration of the modulation, coding, channel capacity and so on. After channel coding and modulation, the SI, MSD and SDI bits will be transmitted on their specified sub-carriers, respectively [8].

1.1 PROBLEM STATEMENT

A. Power consumption measurement in SDR is the key issue in improve the efficiency in the research on the architecture options for industrial embedded architecture design, issues and solutions will be explored to benefit from the data-flow modeling and a simulation tool. The research about design options includes:

B. Describing the SDR receiver use cases, setting up the requirements based on an existed signal processing MATLAB chain and the de-multiplexing sub-system, the statistics about the resource usage on the target platform are also taken as constraints.

C. Building the data-flow model for the SDR receiver. The model should be flexible enough to express the behaviors of the system in multiple transmission modes and spectrum modes for three use cases.

D. Doing simulation with a data-flow simulator using the receiver and several graphs for different architecture design options, in order to generate analyzable data and results.

E. Analyzing and evaluating the designs with the simulation results. The recommended design options can be made for the SDR receiver system.

F. In order to understand the requirements of the processing functions and advanced receiver use cases, it is a task of the project to complete an SDR de-multiplexer, which includes the API design and testing based on an initial SDR de-multiplexer implementation.

1.2 AIM OF CONTRIBUTION

Development of a project management and security parameters in software defined radio-based radiometer. MATLAB based scripts for analyzing data generated by the software defined radiometer using IoT based genetic algorithm. While the modulation scheme in use determines the signal properties that are used to encode data, the modulation order determines the rate at which data may be transmitted. To understand the relationship between constellation size and bits per symbol, the concept of a communications symbol must first be introduced. Rather than transmitting bits, real-world digital transmissions actually transmit these communications symbols. To do this, a constellation that maps the symbols to bit strings must first be constructed. A possible constellation for a Phase-Shift Keying (PSK) modulation scheme with a constellation size of 8. Similarly, a possible constellation for a Quadrature-Amplitude Modulation (QAM) scheme with a constellation size of 16. Real-world SDR modulation schemes are much more complex than this simplistic example but it helps illustrate the main idea. Carrier signal properties in SDR are altered in a controlled way to encode data and the receivers know the scheme by which the data is encoded and can thus recover the data for determining the power consumption.

1.3 SOFTWARE DEFINED RADIO

There are two sub-systems in the SDR receiver, there are two sub-systems: one is de-modulation & de-coding sub-system, in which the SDR baseband signal got from Analog to Digital Converter (ADC) is demodulated and decoded via the baseband signal processing chain. The other sub-system is the de-multiplexing sub-system, which extracts wanted audio and data from the multiplexed frames from the demodulation & decoding sub-system. Two working modes exists in the baseband signal processing chain [9]. The Acquisition & Synchronization mode, the Demodulation & Decoding mode. The receiver always starts from the Acquisition mode when switching to a new channel. After the synchronization has been finished, it will switch to the Demodulation & Decoding mode in this subsystem.

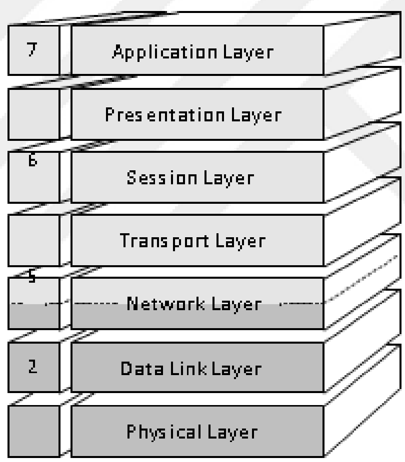


Figure 1.2: The OSI-layer model which depicts the SDR proposition in the first three layers [10].

The SDIs processing includes the de-interleaving task, Viterbi decoding task and SDI descrambling task. De-interleaving task will finish until one logic frame (4 sub-frames) has been received, all processed OFDM symbols accumulate in a buffer until the number is equal with one logic frame. The following tasks, Viterbi decoding and SDI descrambling, work on one logic frame too. A large buffer for the SDIs LLRs remapped from several OFDM symbols is required for the input of deinterleaving task, an output buffer with the same size is also required and it can be released only after the SDI descrambling finished.

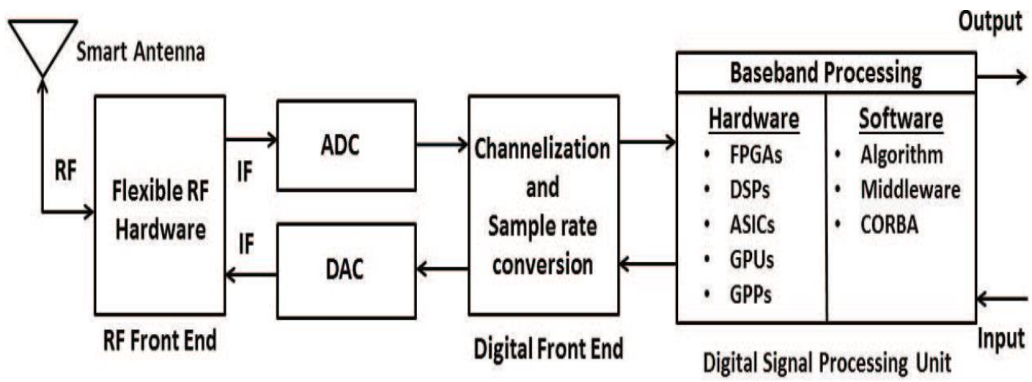


Figure 1.3: The SDR transceiver. Source: [11].

1.4 HIGH-LEVEL AND LOW-LEVEL OPERATION IN SDR

While theoretically any size constellation may be used, there is a practical limit in real transmissions. Given the modulation scheme and constellation, the receiver's job is to map the received signal to one of the modulation symbols in order to determine the transmitted bit string. Due to electromagnetic interference (i.e., noise), there is always some uncertainty in the signal. The uncertainty introduced in the signal may be visualized. In a perfect world, the received symbols would always match the constellation perfectly. In the real-world, however, noise causes the received symbols to be distributed around the constellation point. As a result, the constellation size is constrained by the amount of noise present in the transmission medium and the receiver's sensitivity. If the receiver is not sensitive enough to distinguish between two constellation points, the received signal may be misinterpreted. Similarly, if the interference is high enough, it could cause the receiver to map the signal to the wrong constellation point. In the face of increased noise, then, the constellation size (and thus amount of information that may be sent in one symbol, i.e., data rate) must decrease.

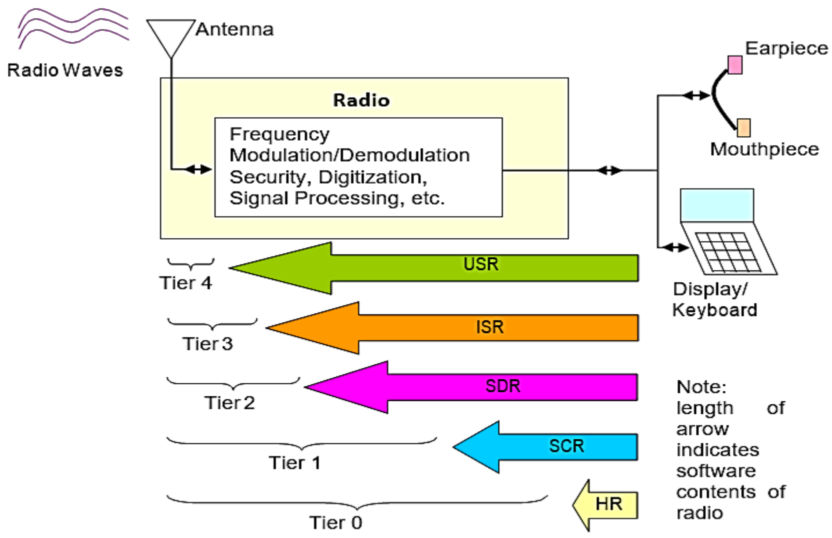


Figure 1.4: The five-tier model of radio equipment that is reprogrammable [12]. (SDR is at the tier 2)

1.4.1 Experiment on MATLAB

In order to interact with the network, users first obtain a CORNET account and then remotely connect to the nodes they wish to use in their experiments via Secure SHell (SSH). In order to run experiments, users must define a master scenario controller and a network topology for each scenario. Then, using MATLAB Software Radio Test System (SRTS) software, the user instructs one SDR node to act as the experiment controller and run the experiment. That node reads the master scenario controller and scenario topology files, initializes the required number of additional SDR nodes and runs the experiment. The results of the experiment are written to log files in the user's directory, ready to be ingested into MATrix LABratory (MATLAB) and analyzed. There are many factors to consider when choosing a simulation package, including

A. Previous experience with one or more simulation frameworks

B. The features offered by each framework and those required by the target network simulation

C. The languages used in the different frameworks and prior experience with these languages

D. Simulation libraries available for the framework and the extent to which necessary functionality is provided in these libraries

E. The extent to which the open-source community has embraced the framework. This will affect the quality and quantity of the available supporting libraries as well as the amount of troubleshooting information that may be found on the internet.

1.4.2 Cognitive Radio

Although the term “cognitive radio networks” is often used to refer to networks in which SUs opportunistically use frequency bands licensed to other PUs, this conception of a cognitive radio network actually falls far short of cognitive networking as originally proposed. The term “cognitive radio” was first coined by [13]. He envisioned future networks in which mobile nodes exchange information about network conditions and their own computational needs to better utilize available resources (e.g., spectrum bandwidth). Using an Artificial Intelligence (AI) component (the “cognitive” in “cognitive network”), these nodes ingest and interpret environmental information far beyond the local transmission environment to estimate and provide for their future resource needs. Additionally, the nodes in the network cooperate to evolve their operating parameters and protocols to improve network efficiency and user experience over time as given in [14].

To help illustrate cognitive networking as originally envisioned, suppose a smart-phone owner, whose device was designed for cognitive networking, uses their phone to purchase tickets for a movie that is showing in two hours at a theater across town as mentioned in [15]. The smartphone would be able to predict that in approximately an hour and half, the user will transit from their current location to the theater. It may also be able to predict the routes the user may take with different probabilities. The device may then communicate this information to surrounding nodes and the supporting network infrastructure. In a fully cognitive network where all nodes are predicting the amount and location of their future resource needs, the network may use this information to optimize the distribution of resources [16]. Interestingly, the environmental awareness necessary to enable these services is already appearing on smart devices. For example, a smartphone running Google’s Calendar application will notify the user when they have an upcoming appointment and will give them a prediction of how it will take them to reach their destination.

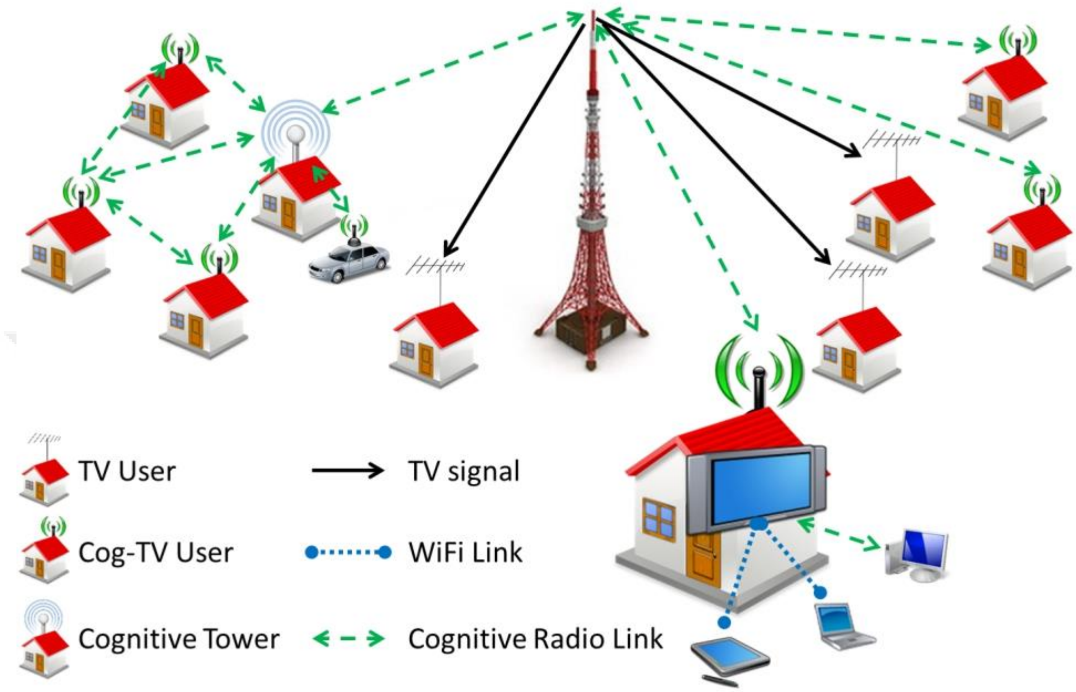


Figure 1.5: The common practice, security and implementation of CR with respect to SDR [17].

2 RELATED WORK

2.1 LITERATURE REVIEW

Due to ongoing fiscal realities within the Department of Defense (DoD), the SDR must continually innovate ways to do more with less. Force readiness and modernization are key issues and the SDR must ensure every dollar is spent with maximum effectiveness as mentioned in [18]. In 2014, the SDR recognized the force's ability to quickly adapt to an operational environment that itself undergoes rapid change as vital to maintaining the nation's strategic advantage in the future. The SDR has further identified the continued development of unmanned and autonomous systems as crucial to the increased agility demanded by the future fight as mentioned in [19]. Two ways in which the SDR is seeking to optimize its spending while maximizing agility is employing its existing systems in new ways and adapting commercial solutions to its needs. By using existing systems in new roles, operators have greater flexibility in ensuring mission accomplishment. Further, effectively adapting existing commercial solutions to its needs allows the SDR to save on development costs while more rapidly fielding required capabilities, allowing the force to quickly respond to emerging threats.

Representative examples of these trends include the use of swarms of smaller Unmanned Aerial Vehicles (UAVs) in missions previously performed by a single, larger platform and the proposal to use Software-Defined Radios (SDRs) to provide intraswarm communications in future missions [20]. The use of UAV swarms offers potential cost savings to the USAF. The vehicles that typically comprise a swarm are smaller and cheaper to build, maintain, operate and replace than their larger coun- terparts. By extension, this reduces operational risk; losing even a handful of smaller UAV platforms is not as costly as the destruction of a larger platform. Further, a UAV swarm can execute new mission types or existing mission types in new ways not possible with a single, larger platform. For example, a UAV swarm could fly in formation to the boundary of an adversary's air defense system, scatter in such a manner as to avoid detection by or overwhelm air defense radar systems and then reassemble itself once the air defense sector has been penetrated as mentioned in [21]. As another example, each platform in a swarm could be outfitted with a sensor suite to ensure continual Intelligence, Surveillance and Reconnaissance (ISR) coverage in a contested environment. If one or more platforms in the swarm is destroyed, the other vehicles ensure continual coverage, a robust mission capability not offered by a single platform performing ISR missions as given in [22].

A UAV swarm that relies on SDRs for intra-swarm communications could provide the basis for a "cognitive" UAV swarm. Such a swarm could adapt its operating parameters to its environment to optimize intra-swarm communication in much the same way nodes in a cognitive network alter their parameters to optimize performance. Before an SDR-based or fully cognitive UAV and UAV swarm may be realized and fielded, however, there exist significant technical challenges that must be overcome as mentioned in [23]. These challenges include the development of a new, or adaptation of an existing, SDR system to a UAV platform. While there are numerous SDR systems available off-the- shelf, the UAV platforms proposed for use in a swarm are typically small platforms. It may be difficult or impossible to retrofit existing SDR systems on board these UAV platforms due to size or weight constraints, especially when the necessary power supplies are taken into account. Perhaps more importantly, a considerable amount of research remains to be done to determine the most efficient and effective algorithms for use in adapting a swarm to its environment as mentioned in [24]. Numerous performance parameters, the dependent optimization objectives and the large number of paradigms available for use in developing such algorithms yield a vast number of possible solutions. Only a small subset of this search space has thus far been explored. It remains to be seen which algorithms will ultimately provide the optimal performance within a cognitive swarm.

2.2 SDR TECHNOLOGIES

In the present research, a cognitive network simulated using the SDR Network Simulator is used as a substitute for a real-world cognitive UAV swarm. While throughput is often the optimization objective of interest, latency is the performance metric under consideration here. Within a cognitive UAV swarm, low-latency is likely to be a critical requirement in intra-swarm communication. In order for the platforms to accomplish the mission, messages sent between them will likely need to be deliv- ered with an upper bound on their delay. For example, if one of the platforms in the swarm is to strike a target, rules of engagement may require the firing platform confirm its location with another swarm member before employing ordinance as mentioned in [25]. If no guarantee can be given on the time frame in which the first platform may expect a response, the opportunity to strike the designated target may be missed. Here, a Genetic Algorithm (GA)-based Cognitive Engine (CE) which alters the modulation scheme and order, transmitter power and bandwidth in an attempt to counter intentional and unintentional interference is developed and tested as mentioned in [26]. The evaluation of the CE will focus on determining whether or not latency may serve as an acceptable performance parameter in tuning the network parameters to optimize performance.

2.3 KEY COMPONENTS OF SDR

Although the term “cognitive radio networks” is often used to refer to networks in which SUs opportunistically use frequency bands licensed to other PUs, this con- ception of a cognitive radio network actually falls far short of cognitive networking as originally proposed. He envisioned future networks in which mobile nodes exchange information about network conditions and their own computational needs to better utilize available resources (e.g., spectrum bandwidth). Using an Artificial Intelligence (AI) component (the “cognitive” in “cognitive network”), these nodes ingest and interpret environmental information far beyond the local transmission environment to estimate and provide for their future resource needs. Additionally, the nodes in the network cooperate to evolve their operating parameters and protocols to improve network efficiency and user experience over time as mentioned in [27].

To help illustrate cognitive networking as originally envisioned, suppose a smart- phone owner, whose device was designed for cognitive networking, uses their phone to purchase tickets for a movie that is showing in two hours at a theater across town. The smartphone would be able to predict that in approximately an hour and half, the user will transit from their current location to the theater as mentioned in [28]. It may also be able to predict the routes the user may take with different probabilities. The device may then communicate this information to surrounding nodes and the supporting network infrastructure. In a fully cognitive network where all nodes are predicting the amount and location of their future resource needs, the network may use this information to optimize the distribution of resources. Interestingly, the environmental awareness necessary to enable these services is already appearing on smart devices as mentioned in [29]. For example, a smartphone running Google’s Calendar application will notify the user when they have an upcoming appointment and will give them a prediction of how it will take them to reach their destination.

As a simple example of how a cognitive network may evolve over time, consider the wireless network shown in Figures 2.2 and 2.3 where Nodes A and B are much closer to the base station than Nodes C and D. As a result, Nodes A and B do not need to transmit with as much power as Nodes C and D to be received at the base station. However, suppose the network administrators initially configure all four nodes to transmit at the same power. As a result, the base station may miss transmissions from Nodes C and D because one or both of Nodes A and B is simultaneously transmitting and the base station cannot hear Nodes C and D over Nodes A and B. If the radio nodes and base station were cognitive nodes, they would be able to communicate about their transmission parameters and their realized performance as mentioned in [30]. Over time, they may discover that decreasing the transmission power of Nodes A and B increases the throughput by allowing Nodes C and D to still be heard when Node A or B is transmitting and thus allow the nodes to transmit simultaneously. (Admittedly, such a scenario would require the nodes to be on different frequencies and the base station to have more than one receiver channel.)

As a result, the network may make the decision to permanently alter the trans- mission power of Nodes A and B. This has the added benefit of making the network more energy efficient; with Nodes A and B using less power to transmit, the network does not require as much energy to run as before. This could be especially important if the power sources on which the nodes run have a finite amount of energy (e.g., batteries) before they must be replaced or recharged. In this hypothetical scenario, neither the increased throughput nor the energy savings would be possible without cognitive networking as mentioned in [31].

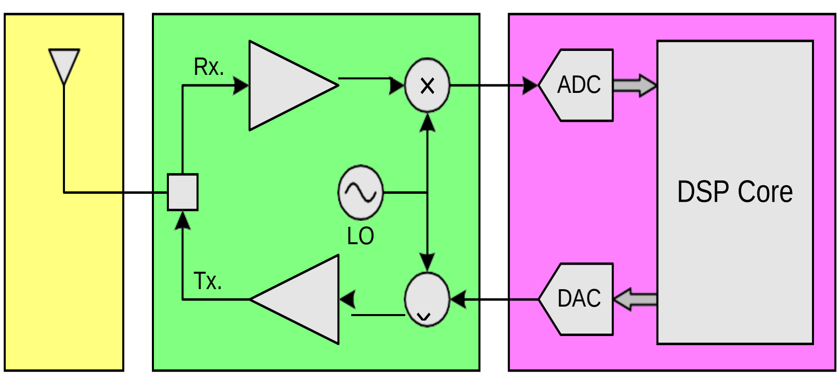


Figure 2.1: Depicts the main components of the SDR [31].

As stated previously, cognitive radios and networks as designed and implemented to this point have fallen considerably short of Mitola's original vision. Most of the net- works realized to date have focused on conditions in the local operating environment to inform decisions as to how to alter operating parameters to optimize performance metrics of interest. The scope of information potentially pertinent to the network in these research efforts is thus reduced in comparison to a fully cognitive network as mentioned in [32]. Cognitive networking as originally envisioned is ambitious and a full realization of such a network from the start would be an impossible undertaking. The ongoing research involving cognitive radios and networks is laying the foundation for the full realization of cognitive networking. Two vital components that have emerged from these research efforts are SDRs and Cognitive Engines (CEs).

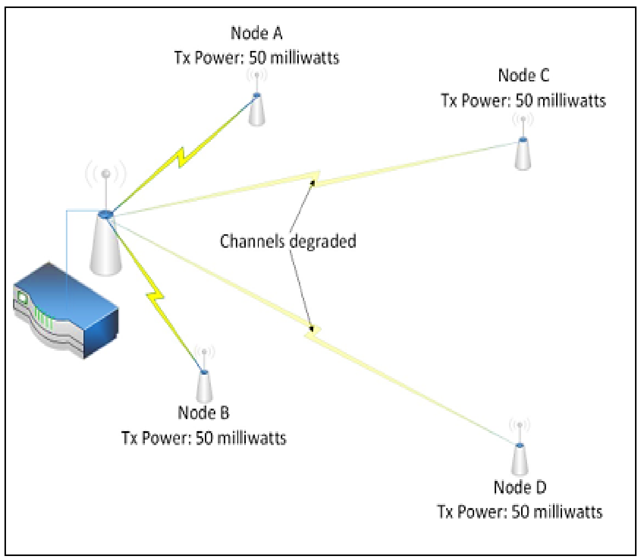


Figure 2.2: Initial conditions in cognitive radio network before nodes perform any information exchange, parameter adjustments or algorithm space exploration [32].

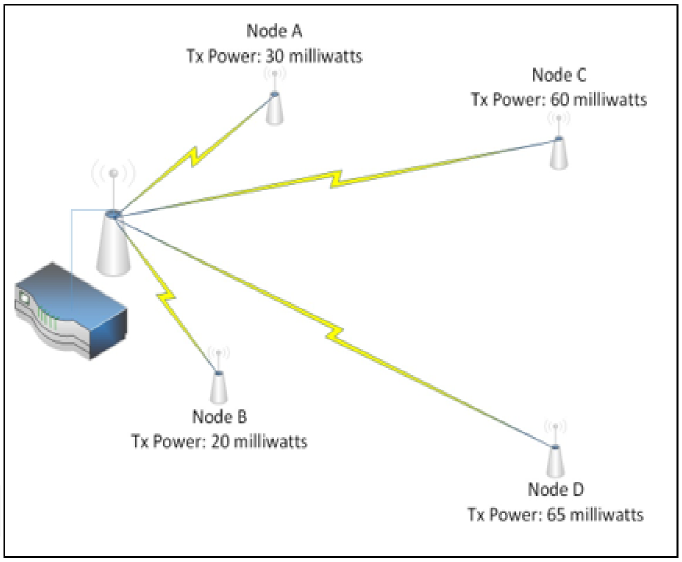


Figure 2.3: Cognitive radio network after parameter adjustment [32].

2.4 RADIO FREQUENCY LINEARIZATION AND SDR

There are also numerous proposed CEs that combine techniques or algorithms to create hybrid engines wherein the performance using a combination of approaches is better than what any individual approach offers. A hybrid CE for achieving efficient spectrum management in cognitive networks is detailed in [33]. Determining priority amongst competing SUs is handled by a new queuing scheme and SUs share information with each other as they are added to the network rather than each evolving its behavior in isolation. In [34], a hybrid engine based on Case-Based Reasoning (CBR) and Decision Trees (DTs) is presented. The CBR path in the engine is used when environmental conditions are within a pre-defined bound to a case previously encountered, defined and stored in the engine's database while the engine's DT path is used when operating conditions not previously encountered are observed. In this case, a new case is defined and stored in the engine's database. A CE based on a hybrid Case-Based Reasoning-Quantum Genetic Algorithm (CBR-QGA) is proposed in [35]. Rather than random population initialization as if often done in GAs, the engine uses case-based reasoning to initialize the population with solutions that are expected to perform well based on previously encountered conditions and the solutions that worked well in those situations. Then, the Quantum Genetic Algorithm (QGA) GA evolves new solutions from the population initialized via CBR. Tests dis- cussed in the paper indicate the new engine outperforms existing CBR-, GA- and QGA-based engines in terms of converge speed and quality of optimization.

Several other AI approaches that have been or could be used as a basis for CEs are explored in [36]. The paper discusses factors that may affect the selection of a given technique for a CE depending on the application and the ramifications the selected AI approach will have on the engine itself.

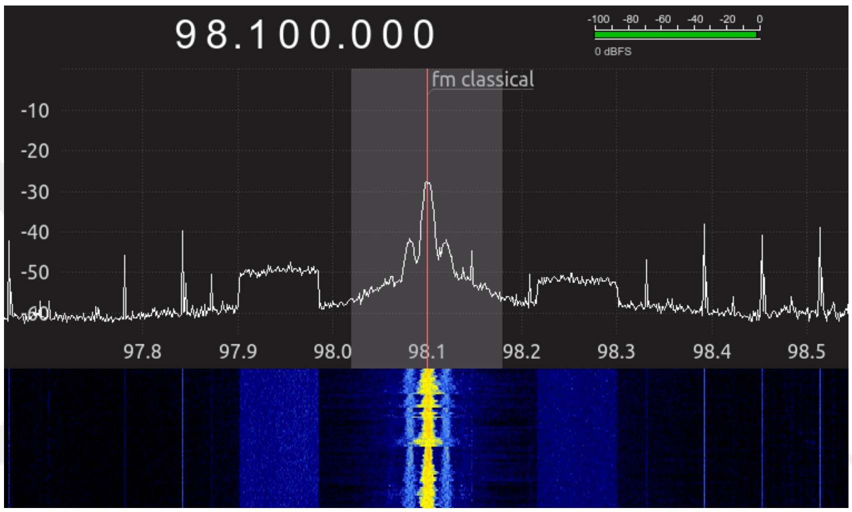


Figure 2.4: Radio frequency linearization through the software defined radion mechanism [36].

Two different problems from graph theory are used here to illustrate these two types of problems as well as introduce GAs. In general, a graph is a set of nodes N and a set of edges E, where each edge e ∈ E originates at a node n1 ∈ N and terminates at another node n2 ∈ N. Further, a weight wn may be assigned to each edge en ∈ E, where n ≤ |E|. Graphs are often used to describe some sort of relationship between nodes, with the edges connecting the nodes providing information about the relationship between the nodes it connects. For example, a graph showing cities with direct flights between them and the cost of each flight is shown in Figure 2.4. This graph may be used to find the shortest distances between two cities, where "distance" is defined according to the weight assigned to each edge. Thus, as the weights assigned here are flight costs, the graph may be used to find the cheapest flight(s) between two cities. Polynomial-time algorithms, such as Dijkstra's Algorithm, are known that provide provably optimal solutions for this problem as mentioned in [37].

Another fundamental graph theory problem, arguably more interesting because it belongs to the NP class, is the infamous Traveling-Salesman Problem. Put simply, given a graph, the Traveling Salesman Problem seeks to find the optimal (usually, "shortest" by some metric) route such that the salesman visits all nodes, normally conceptualized as cities, exactly once and returns to the node (city) from which they started [38]. For example, given the graph in, the Traveling Salesman may wish to find the least-costly route, starting and ending in SDR, such that all other cities are visited exactly once.

This may seem trivial at first glance and certainly for a small graph such as the finding the optimal solution is relatively easy. However, as the number of edges and nodes in the graph increases, finding the optimal solution becomes more difficult. For example, if the graph included all flights for all airlines between all cities in the United States, finding the optimal solution the Traveling Salesman Problem would be much more difficult. Finding the solution to this problem involves considering all possible paths of length that originate and terminate at a selected node in the graph and visit all other nodes. Then, for each such path, it must be determined whether the path provides a route from the desired starting city to all other cities and then back to the originating city and does so more cheaply than all other routes. To obtain the correct number of possible paths and avoid double-counting paths, (2) must be divided by two to obtain the correct number of possible paths as given in [39]. Therefore, in the graph as the number of flights (edges) grows, the number of possible solutions grows exponentially. As a result, with even a relatively small number of possible flights, the Traveling Salesman Problem becomes intractable.

Rather than simply giving up on NP problems, computer scientists have designed numerous algorithmic paradigms to attempt to solve such problems. Rather than looking to optimally solve instances of these problems, so-called approximation approaches seek to find "good enough" solutions, with some algorithms developed from these techniques being able to guarantee a solution within some bound of the optimal solution. One class of approximation algorithms, to which GAs belong, are referred to bio-inspired because they are modeled after natural phenomena.

2.5 GENERIC SYSTEM SPECIFICATIONS OF SDR

The power consumption problem will be used to illustrate the GA paradigm. The first step is to determine how solutions to the problem may be encoded. As solutions consist of different combinations of edges from the edge set, they may be specified as bit strings of length |E|. Then, after sequentially numbering the edges in the bit position corresponding to a given edge indicates the edge's inclusion in a proposed solution whereas a 0 indicates the edge is not included in the proposed solution. Given this encoding, the chromosome encodes a possible solution consisting of edges 1, 2, 4, 10, 11, 16, 17 and 18. Now that an encoding scheme is established, the next most important part of the algorithm, the fitness function, must be developed. While here the problem itself suggests a fitness function, it may become quite complex depending on the

Figure 2.5: The ultra-electronics to develop SDR radios for US Army communications [40].

For the power consumption problem, the fitness function must assess two things. First, it must total the cost of all flights in the proposed solution. Finally, the mutation mechanism, which affects the algorithm's ability to break out of local search and explore other regions of the search space, must be determined. General recommendations as to the population size and number of generations to use are available and several different mutation mechanisms have been proposed [41]. In general, the number of existing solutions selected for recombination, the recombination mechanism itself and the number of new population members thus produced are arbitrary and may be specified by the algorithm's designer, though two solutions are often selected for recombination into a single new solution [42].

Two solutions to the Traveling Salesman Problem are used to illustrate the GA's execution. The first solution is the same as given previously while the second is another valid solution to the problem. (A randomly-initialized population is assumed here and not included in full; it is further assumed these two solutions were selected at random from this population.) Once the algorithm chooses these two solutions, it must recombine them in some way to form a new solution. Here, single-point crossover in which the chromosomes are split into two halves and opposite halves recombined to form a new solution, is used. The two solutions are split after the eleventh bit and then the first half of the first chromosome and the second half of the second chromosome are used to form the new population member. After the new chromosome is formed via recombination, the new population member must be selectively mutated.

2.6 SDR ARCHITECTURE

After mutation, the fitness function is evaluated on the new population member and, if the new member's fitness value is high enough, it is added to the population. After the new member is selectively added to the population, the entire process begins again with two new members of the population being selected for recombination. This process repeats for however many generations the algorithm is programmed to use in finding "good" solutions. When this number of generations is reached, the algorithm returns the best solution it has found.

As stated previously, several factors such as the population size and number of generations the algorithm is set to use in finding solutions affect its performance. Another important measure related to signal power is signal-to-noise ratio (SNR). As its name implies, this is the average signal to average noise power and is often reported in dB. Similar to dBm, dB is a ratio between two signals, except with dB, the signal to which a signal is compared is arbitrary, not 1 milliwatt [43]. Therefore, as SNR increases, the signal power becomes increasingly stronger than the noise power. Bandwidth refers to the frequency range across which a given signal is transmitted and relates to the amount of information the signal may carry at a specific point in time. In general, a signal with a wider bandwidth may carry more information. Increasing the bandwidth without similarly adjusting the symbol rate simply leads to more samples for the same symbol rather than additional symbols (information) [43]. FEC may be thought of as adding information to the transmitted signal to combat signal degradation and data loss. With FEC, the transmitter pre-processes the bits to be sent before modulation and adds additional bits to the data. On the receiving end, the receiver uses this additional information to detect and correct bit errors that occurred during transmission. The manner in which the transmitter and receiver pro- cess the signal depends on the encoding scheme used and how robust the transmission is against bit errors varies between FEC techniques. Additionally, no FEC scheme can allow for all bit errors regardless of interference level in the transmission medium to be corrected.

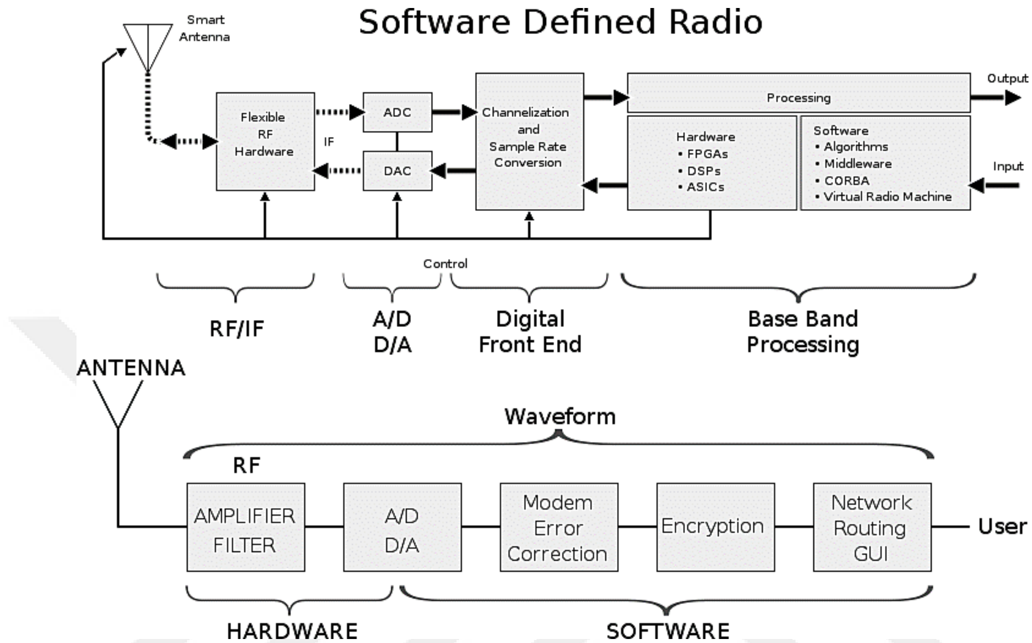


Figure 2.6: The basic architecture of SDR and framework which takes signals from the antenna and directly forwards to the user through waveform [43].

Modulation scheme and order refer to the manner in which data to be transmitted is encoded on a carrier signal. In general, carrier signal properties (e.g., frequency, amplitude, phase, etc.) must be altered, or modulated, in such a way that their values, or the difference in their values in successive signal samples, may be interpreted in accordance with an agreed upon scheme and the data extracted.

.

3 METHODOLOGY

The research in determining the set of SDR parameters is deciding on valid values for each variable. While a large search space is desirable, it is possible for the search space to grow too large. In such cases, a genetic algorithm would either need an inordinate amount of time to search a reasonably-sized portion of the search space to increase confidence it finds a good solution or it would search a portion of the search space that is too small and which yields low confidence it finds a reasonably good solution.

3.1 POWER CONSUMPTION REQUIREMENTS

Similar to the OMNeT++ simulations, another independent variable of interest here is the number of generations used by the engine and its effect on the quality of the solutions the engine returns. As discussed above, due to the manner in which the MATLAB engine reimplementation estimates BER, the number of generations is expected to have a significant impact on the engine's runtime. If this is indeed the case, as with the previous experiments, determining the point at which the increase in runtime does not yield enough improvement in the fitness of returned solutions to warrant using additional generations in finding them becomes an important part of future work. Additionally, the runtime of the MATLAB engine implementation may be compared to the C++ engine implementation included in the SDR network to comment on the importance of careful engine design due to the effect implementation decisions (e.g., coding language, data structures, variable declara- tions and calculation approaches, etc.) may have on the runtime of an actual engine included in a real-world Unmanned Aerial Vehicle (UAV) swarm.

The next four independent variables of interest are the four tuning knobs made available to the engine: modulation scheme, modulation order, power and bandwidth. Due to the small size of the search space used in the engine developed here, it may be exhaustively searched to find the optimum solution. As each variable deviates from its optimum setting, the fitness of the resulting solution will decrease. However, the rate at which this deviation from optimum fitness occurs is of interest. To explore this, the resulting fitness values as each parameter varies through its entire range of settings while the other values remain constant at their optimum settings will be considered.

3.2 MAPPING IOT ENCODING TO A SOFTWARE DEFINED RADIO

It is expected IoT encoding will cause an increase in the engine's performance, evidenced by the engine returning solutions with lower BERs and higher latency fitness than those returned for the same non-IoT encoded case. However, it is also expected IoT encoding will increase the runtime as compared to the non-IoT encoded case. Further, this effect is expected to be more pronounced as the noise floor becomes closer to the transmit power (i.e., as the Signal-to-Noise Ratio (SNR) drops).

3.2.1 Power Consumption in SDR

Constraining the modulation schemes under consideration is relatively easy. The SDR modules in use here only support Quadrature Amplitude Modulation (QAM) and Phase Shift Keying (PSK) modulation. For constellation size, the SDR modules in use allow for powers of two up to and including 256. Such a large con- stellation size is virtually impossible to achieve with PSK modulation and difficult to achieve with QAM except in scenarios with extremely low noise levels.

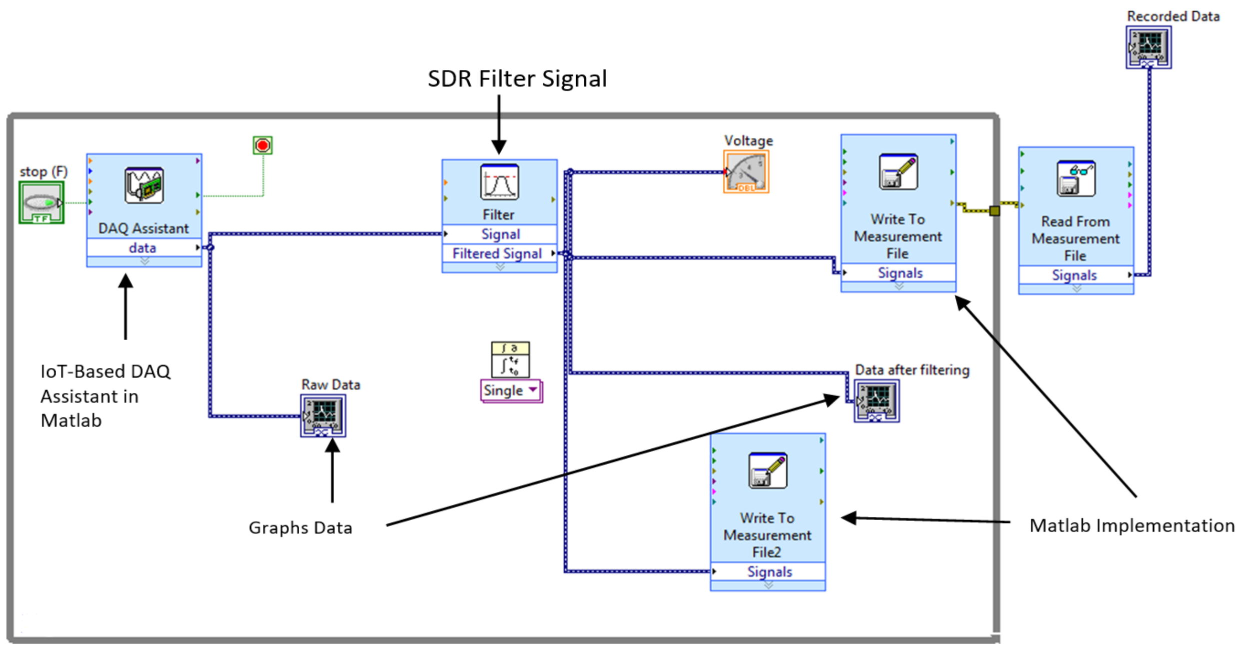


Figure 3.1: A set of procedures that is being followed is represented by the pictorial representation of flowchart.

3.2.2 Data smoothing

Pure Software Radio is yet a distant data smoothing due to high architectural Complexity and high Flexibility. Software Defined Radio(SDR) is the one which has medium architectural Complexity and more than medium Flexibility. These radios are reconfigurable one where in the software can be upgraded or added with other modes of radio or frequency of operation. A typical SDR development kit with all front panel interfaces. This has USB, Ethernet and HDMI Interfaces for interacting and viewing the performance, with RF Interfaces for connecting the antennae. The schematic diagram of such Universal Software Radio Peripheral(USRP). The Software Development Kit module such as GNU Radio can be used to configure the functions.

3.3 SOFTWARE DEFINED RADIO MANAGEMENT

Software Defined Radio Management is performed by five components. Domain Manager, Application Factory, Application, Device Manager and File Manager. Domain manager is the central component for radio management. It accepts registration of every Device Manager, Device, and service. It is responsible for the installation of application and creates an Application Factory. It offers introspection services such as listen to events, get the list of installed and deployed applications. It provides two event Channels, viz., one for every device state changes and another one for radio management events. Domain Manager creates an Application Factory for each type of installed application.

3.3.1 Software Defined Radio Management Tools

SDR can be developed using various combination of Universal Software Radio Peripheral (USRP) and Radio Software Toolkit. Few open source and commercial Toolkit are listed below:-

GNU Radio is an open source toolkit of signal processing blocks which are written in MATLAB for implementing SDRs on external /RF hardware. GNU Radio Companion (GRC) provides GUI that allows flow graphs which are built in MATLAB and signal visualisers to be built out of signal processing blocks. USRP could be from National Instruments to cover frequency range from HF to UHF ranges.

Prismtech has introduced a product Spectra CX bundle and alongwith Mathworks' Simulink for the development of SCA compliant SDR. Spectra SDR is useful in designing and simulating waveforms, transforming the algorithmic blocks into deployable components, deploy and test Waveforms and applications.

National Instruments [37] Labview software and USRP will be another toolkit for development of SDR.

SCARI software suite from the Communication Research Centre(CRC) Canada is a high performance SDR waveform development software and debugging toolkit.

Pentek company produces various SDR modules for further deployment in various frequency bands and other available SDRs and SCA Operating Environment and Development Tools.

Also some Simulation Package such as SciLab, GNU Radio or MatLab can be used for verifying the SDR. The performance can be verified with parameters such as Signal to Noise Ratio (SNR), Bit Error Rate (BER), Mean opinion Score (MoS), Spectrum Visualisation and Log Spectral Distance (LSD).

3.4 SDR MANAGEMENT THROUGH SOFTWARE COMMUNICATION ARCHITECTURE

Over a period of decade, the SDR designer and developer have started agreeing to a unified architecture called Software Communication Architecture(SCA). This was initially developed at the instance. At present SDR version 4.0.1 was accepted as the last known SCA Specification standards and released in the public domain on 18 Nov 2013 for development of SDR. The present Radios are Hard-wired and works only in narrow-band and for few fixed modes of Operation only. Thus any feature upgrade makes the Ownership cost very high. But, by implementing Radios in a Software programmable manner. The existing Hardware Radio cannot be remotely repairable and makes the maintenance cost very high. With the software Defined Radio adhering to SCA standards will have access anywhere with authentication and remote repair and upgrade is feasible with experts centrally located anywhere in the Globe. The software radio configuration for different modes can be developed and loaded into SDR in the form of Waveforms. Thus friendly countries can have common Waveforms and communicate efficiently with single Software Radio. This will be possible with the advent of advanced Digital Signal Processing(DSP) techniques at Software level running in either General Purpose Processor(GPP) or DSP Chip or Field Programmable Gate Array(FPGA). SDRs have the ability to work beyond simple single-channel, single-mode transreceiver technology, by accommodating arbitrary mode-changes because of the flexibility in determining the channel bandwidth, data rate, and modulation.

This is an essential part of the SRM where SDR is deployed. This section assists the SDR to maintain the integrity of RA by making it appear as hardware so that the OS knows that it is execute-only. The security implemented in this section does not allow the OS kernel into memory in this segment. The OS interact with the V-HW via a controlled interface. This segment combines the software component with hardware characteristics to ensure security.

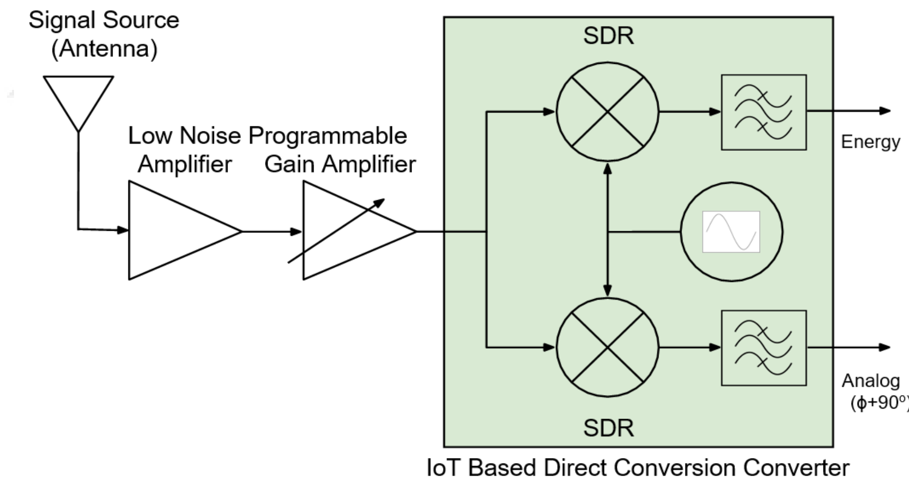
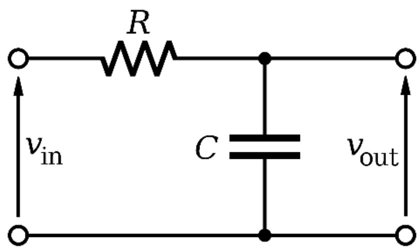


Figure 3.2: The relation of the middleware and the OS.

The SDR which are compliant to SCA, are remotely accessible for configuration, loading of new waveforms and bug-fixing. Thus new products can be introduced by remote downloads. This saves the operating cost, development cost, capital cost in new product introduction. TCP/IP security and performance are key requirements of SDR Devices. The SCA is characterized by a particular combination of software components and connections. This SDR Standards or framework enables porting of “Waveforms”, reuse of software, extensibility of HW and SW, interoperability between platforms and use of commercial product line. Since SDR is in essence a radio whose output is determined by software, the output is entirely reconfigurable at any given time.



V_out= V_in^2=nP_in

Figure 3.3: A simple RC circuit for voltage.

3.4.1 Universal Software’s for Management Of SDR

the performance objectives of interest and parameters the engine may use to tune network performance are established, the function that will be used to produce a fitness value for each solution considered must be developed. Rather than developing a single function to capture all performance objectives in a multi-objective problem, it is usually easier to develop separate fitness functions for each objective and then combine them together. This approach will be used here. In order to accomplish this, care must be taken to ensure the possible range of values output by each of the individual fitness functions is the same. This is often accomplished by normalizing the functions to output a value between 0 and 1. Finally, it is also easier if all objectives have the same goal (i.e., minimization or maximization). Due to the manner in which the individual objectives are normalized here, discussed below, latency and throughput fitness value close to 1 are preferable while power fitness values close to 0 are preferable. Before the power fitness may be added to the fitness value for the other two objectives, it must be transformed into a maximization problem. Otherwise, tradeoffs in latency or throughput and power consumption may mask which of two solutions is preferable. Thankfully, as the possible values that may be returned by each of the fitness functions is normalized to the range of 0 to 1, this transformation may be easily accomplished by simply subtracting the fitness value returned for the power objective from 1. In that way, the goal of minimizing power consumption is the same as maximizing the resulting value when the normalized power fitness value is subtracted from 1. With the fitness function structure as described, solutions with larger fitness values are preferable and different performance objectives may be given different levels of importance by incorporating weights into the fitness function.

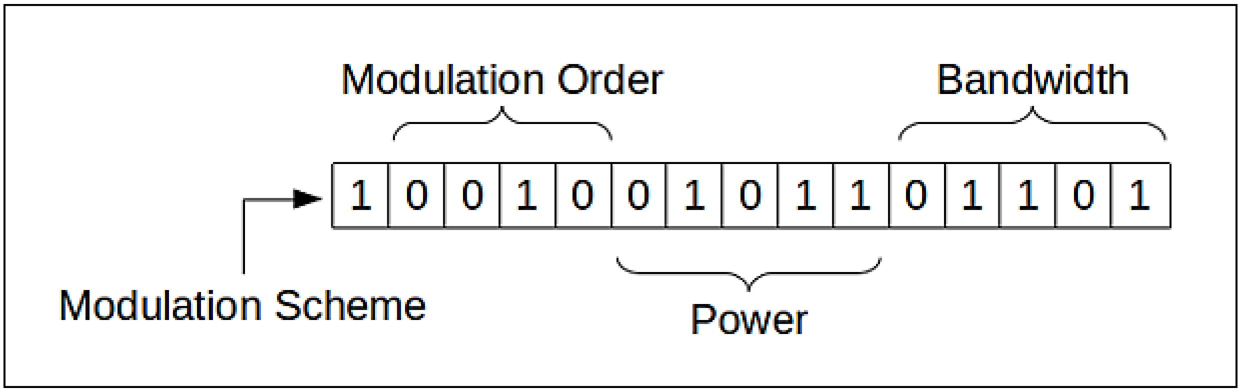


Figure 3.4: Flow diagram of an SDR with the distinction between the digital and analog components.

The SDR bandwidth, carrier power and modulation order are three of the variables the CE has at its disposal to optimize network performance so their values will be known when trying to evaluate the fitness function. Similarly, the noise power may be obtained from the radio medium module within the simulation. The symbol period is a function of the bitrate, which is set at the beginning of the simulation and is not allowed to vary, and the modulation order. In particular, dividing the bitrate by the modulation order gives the symbol rate. Then, taking the inverse of this value yields the symbol period. Thus, all the necessary values are known when trying to evaluate the fitness function.

The more difficult step in determining the bit error rate when trying to calculate fitness values is the evaluation of the SDR function. The integral has no closed-formed solution. While it could be numerically evaluated, the usual approach is to use a pre-compiled to look up the approximate value. This is the approach used here.

3.4.2 Genetic Algorithm on SRD

The SRD may be seen that for the latency and throughput objectives, lower BERs correspond to more perfect reception that closely matches the theoreti- cal performance achieved in the absence of bit errors. As this is the value by which the realized latency and throughput are normalized, fitness values close to 1 indicate performance that more closely matches this limit while fitness values close to 0 indicate performance that is considerably below this upper bound. Thus, latency and throughput fitness values close to 1 are preferable. Similarly, for power fitness, before subtraction from 1, values close to 1 indicate power consumption that is approaching the maximum possible power consumption while values close to 0 indicate little power consumption. Thus, maximizing the value that is obtained when the power fitness value is subtracted from 1 is the same as minimizing the power fitness value and, by extension, the power consumption.

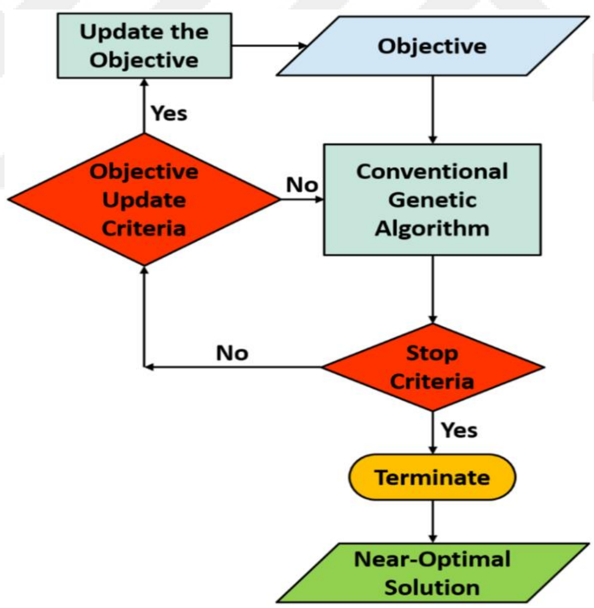


Figure 3.5: The flowchart of genetic algorithm depicts multiple layers architecture of techniques for handling the electronic platform in the radio processing system (RPS).

From flowchart of genetic algorithm, the required inputs for the engine's genetic algorithm include the current population and the number of generations the engine should use in pro- ducing new solutions. Once the engine produces the number of specified generations, it returns the most-fit solution it has found. In generating a new population mem- ber, the engine randomly selects two existing population members. The engine then splits these members into their constituent genes. For each of the new population member's genes, the engine randomly selects one of the two genes contributed by the parent population members and assigns it to the new population member. Once the engine forms the new population member, it mutates the newly formed chromosome. For modulation scheme, power and bandwidth, this mutation is done randomly such that, on average, one bit in each chromosome is mutated. Due to restrictions on the modulation order, mutating this gene is not handled in the same fashion. First, if the mutated modulation scheme is QAM, the mutated modulation order must be even. Further, regardless of the mutated modulation scheme, the mutated order cannot exceed 8 (a binary string consisting of a 1 followed by three 0s). Therefore, if the mutated first bit is a 1, it sets all other bits. Thus, the mutation is handled such that, on average, the bit only mutates one out of every eight times. The remaining bits are mutated in a similar fashion as the bits for the other genes. Once the engine mutates the modulation order, a final check to ensure QAM and an odd modulation order are not combined in the new population member. If the engine detects this situation, it sets the final bit of the modulation order gene to 0 and randomly selects one of the other 3 bits to set to 1.

3.5 IOT-BASED DAQ ASSISTANT AND GRAPH DATA

In exploring a cognitive network's performance with a genetic algorithm-based engine that targets latency for optimization, a lower bound on the latency of exchanged messages in a wireless network to which the cognitive network's performance may be measured is needed. Though these tests could be run with a set configuration (e.g., a single modulation scheme, constellation size and data rate), the results may only be valid for that particular configuration. It is expected latency will be largely independent of such factors as the major contributors to latency are processing steps that must occur in all transmitters and receivers. However, to ensure no unanticipated factors cause the latency lower bound determined through this experiment to fluctuate significantly across configurations, several combinations of modulation scheme, constellation size and data rate will be tested.

One of the most powerful aspects of the SDR framework is the considerable number of transmission parameters it allows users to vary during the course of their experiments. The

Cognitive Radio Test System (CRTS) allows the user to specify over 50 different transmission parameters in a given scenario, including transmitter and receiver modulation scheme, constellation size, data rate, frequency, gain, subcarriers and Forward Error Correction (FEC). Additionally, a user-provided Cognitive Engine (CE) may alter any of these parameters as deemed necessary during the scenario. As the number of parameters that may be altered, as well as the number of values each parameter may take on, lead to a number of test cases that exhibits exponential growth, the first step in the design of the SDR experiments is to identify a subset of the parameters to be investigated (independent variables) and the resulting system response indicators (dependent variables) of interest. As previously identified, the network performance metric (dependent variable) of interest here is the latency with which network-level packets are sent and received between nodes.

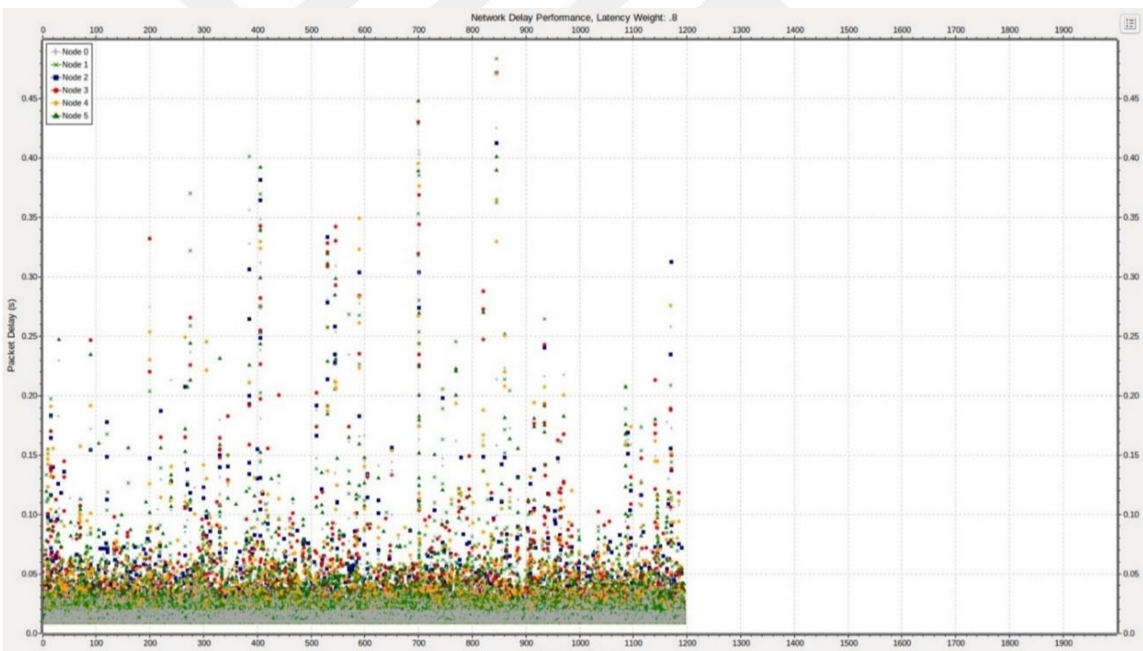


Figure 3.6: Illustration of network delay secure technique for mobile terminal using SDR.

The independent variables under investigation are modulation scheme, constella- tion size and data rate. Within DAQ, the modulation scheme and bits per symbol are controlled by the same parameter. The modulation scheme affects the efficiency with which data is encoded and thus the rate at which data is transmitted from sender to receiver. It is expected latency with Quadrature

Amplitude Modulation (QAM) will be less than with Ampitude Shift Keying (ASK), which is expected to be less than the latency with Phase Shift Keying (PSK) due to a decrease in the encoding efficiencies across the three modulation schemes. Further, it is expected that as the constellation size used in conjunction with a given modulation scheme increases, the latency will decrease. The constellation size increases, the number of bits that may be sent simultaneously also increases. Therefore, with a larger constellation size, information may be sent at a faster rate and the time required for the sender to transmit a given packet to the receiver is less than the time required with a smaller constellation size. Finally, it is expected the data rate will likely increase latency. As the data rate increases, the radios are exchanging messages at a faster rate, resulting in a greater number of collisions. This will likely require packets to be retransmitted, resulting in greater latency. This research shows the independent variables of interest, the values for each variable to be investigated during the course of the experiments and the CRTS parameter that controls each variable.

3.6 DIGITAL SIGNAL PROCESSING IN SDRS

There are a number of important assumptions made in establishing the experi- mental design, including:

To decrease unintentional interference, the same two adjacent SDR nodes are used throughout the experiments. While this may not entirely eliminate interference, it is assumed this design choice reduces the strength and duration of any electromagnetic interference with respect to the transmission power and length of exchanged messages such that any residual noise the transmissions do encounter will not appreciably increase latency.

SDR only supports a limited number of modulation schemes by default. While other, more advanced schemes may offer slightly better performance, it is assumed the latency bound determined with the modulation schemes available by default will not differ significantly from that found using a more advanced scheme. Additionally, the schemes investigated here are standard schemes, used in a wide range of applications whereas more advanced schemes may only be applicable to a small number of more advanced scenarios.

Messages will be automatically generated by the SDR nodes. It is assumed they will exhibit sufficient randomness as to approximate normal network traffic such that the latencies observed during the SDR experiments will not differ significantly from real-world network traffic.

Latency measurements will be performed using native SDR network layer performance metric reporting capabilities and it is assumed these capabilities are of sufficient granularity as to support meaningful latency measurements.

Node synchronization will be handled automatically by SDR via the Network Time Protocol (NTP) and it is assumed the synchronization is of sufficient accuracy as to not invalidate latency measurements.

The SDR experiments will test the performance of each possible combination of modulation scheme, constellation size and data rate. Before executing any test cases, the necessary CRTS configuration files will be created and pre-staged in the CORNET file system. In order to obtain a sufficient number of samples such that the genetic algorithm holds, each test case will run for 200 seconds. The latencies of all packets sent from one node and received at the other node during each test case run will be averaged together to determine the latency figure for that case. Additionally, to obtain a single per-packet latency value against which the network's performance with a cognitive engine may be compared, the latency values determined for each test case will be averaged together. The variance for each test case as well as for the latency figure determined by averaging all cases together will be evaluated to determine if the latency values so determined are valid. If not, multiple latency values, representative of different network and node configurations, will be needed and the cognitive network's performance with each of these configurations will have to be compared against the associated latency value individually.

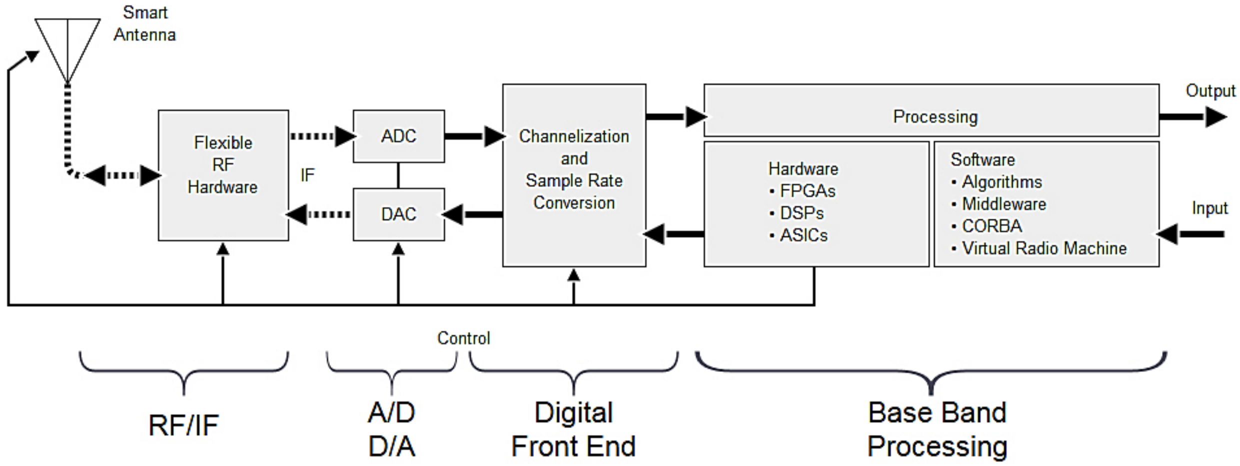


Figure 3.7: Depicts SDR techniques and flexibilities in the hardware architecture.

The SDR nodes used in the experiments must be carefully chosen so as to not introduce additional, unnecessary sources of latency. The selection of the two communicating nodes must ensure they are adjacent to minimize the amount of latency caused by propagation. (If necessary, this could be removed from the data by calculating how long message propagation takes and subtracting this value from all observed data points.) Further, the same two nodes should be used for each test case. Conversely, the selection of the master node, which coordinates each test, is not subject to any constraints. As the master node does not directly take part in the test (i.e., does not create or exchange any messages whose latency will be measured), its location in the network with respect to the other nodes is of no consequence to the latencies observed.

Table 3.1: Commercially available SDRs.

SDR's Nodes	IoT Encoded?	Data Rate	Bandwidth	Power	Price
1	No	900 MSps	32 bit/sec	0.55 W	$25
2	Yes	1.4 GSps	64 bit/sec	1.22 W	$150
3	No	560 MSps	32 bit/sec	0.56 W	$55
4	Yes	2.2 GSps	64 bit/sec	1.08 W	$130
5	No	300 MSps	32 bit/sec	0.78 W	$30

3.7 SECURITY IMPLEMENTATION PROCESS OF ELECTRONIC PLATFORMS

To run the tests, the SDR software is invoked on the master node and supplied with a master configuration file specifying which scenarios to run. The node then initializes the two communicating nodes with the operating parameters specified in the supplied configuration file for the first scenario. The two communicating nodes then exchange messages and record their timestamps for 200 seconds. At the conclusion of each test, results are automatically written into log files and converted to Octave scripts for MATLAB ingest. Once the first test concludes, the master node reinitializes the communicating nodes with the operating parameters for the second scenario. The nodes then again exchange messages and record results for 200 seconds, writing the data to a log file when the test concludes. This process repeats for all scenarios specified in the master configuration file.

SDR automatically provides a vector of message identification numbers and the timestamps at which the transmitter sends those messages and analogous vectors for the receiver which contain the message identification numbers and the timestamps at which those messages were received. Thus, to measure the latency of a given re- ceived message, the timestamp at which it was sent from the transmitter will be subtracted from the timestamp at which it was received at the receiver to obtain a latency measurement for that message. Care must be taken to ensure the timestamps extracted from the two vectors for use in this calculation are for the same message. It is expected that not all messages sent will be received so simply subtracting the two vectors will likely over-estimate many messages' latency by comparing unrelated timestamps. Instead, the indices at which the same message identifier is found in the two message identification number vectors will be used to index into the timestamp vectors and extract the timestamps at which each message was sent and received.

4 RESULTS

For these experiments, the independent variables of interest are latency weight, noise floor and number of generations. The latency weight indicates the priority given to the latency component in the fitness function. As it increases, solutions that emphasize lower Bit Error Rates (BERs) to improve latency performance are assigned higher fitness values than solutions that trade some latency (and throughput) performance to reduce power consumption. Therefore, as the latency weight changes, it is expected the solutions returned by the engine will change and the latencies realized in the network will decrease. As a result, it is expected the cognitive network will have increasingly improved latency performance as compared to its non-cognitive counterpart as the latency weight increases. As the number of generations the engine uses to develop solutions increases, it potentially allows the engine to search previously unexplored regions of the search space (through global search) or find better solutions in the vicinity of previously found "good" solutions (through local search). It is expected, then, that the fitness of solutions returned by engine as the number of generations it uses to evolve solutions increases will, in general, improve. Thus, as the number of generations the engine uses grows, it is expected the network's latency performance will continually increase as compared to the non-cognitive, baseline network. Rather than having a node in the network to produce interference, the back- ground noise level is incremented throughout the test runs to simulate the effect of a broadband jammer. It is assumed varying the background noise level in this manner has a similar effect on the network nodes and their ability to communicate as a broadband jammer periodically increasing its output power would have. As with SDR, genetic algorithm only supports a limited number of modulation schemes by default. Here, only PSK and QAM are considered. While other, more advanced schemes may offer slightly better performance, it is assumed the trends noted in the network's latency performance with and without the engine will not differ significantly from that found using a more advanced scheme. Beyond this, it is assumed the traffic characteristics (e.g., inter-packet delays, etc.) are sufficiently random as to approximate real-world network traffic.

Additionally, it is also expected that as the number of generations increases, the engine's runtime will also increase. The rate at which this increase occurs is of interest. Here, the engine has arbitrarily been set to run every 5 seconds. However, depending on how fast conditions in the radio medium are expected to change, this parameter could be set to a wide range of values. Therefore, how many generations may be developed, and thus the portion of the search space that may be explored in a given amount of time, is of interest. We built the system with MATLAB.

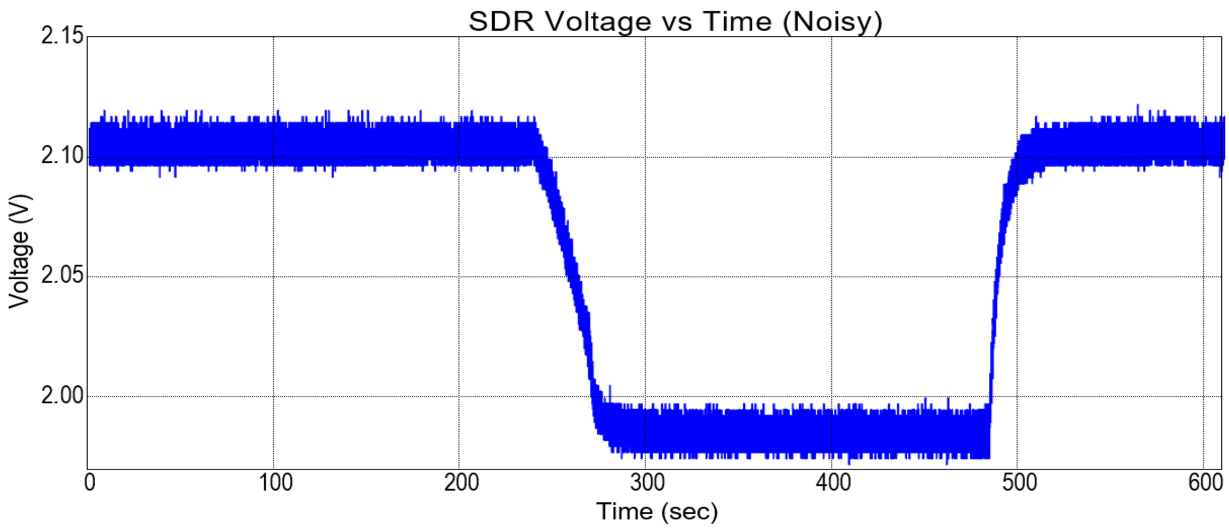


Figure 4.1: The evaluation of minimum voltage with respect to time for low energy consumption with IoT filtering and noisy signals.

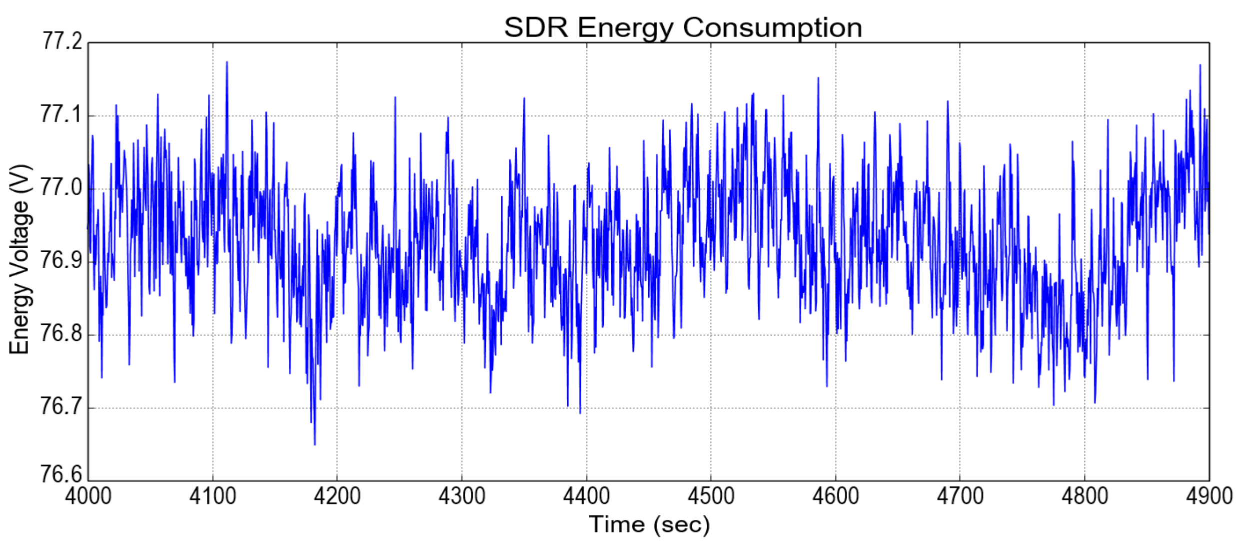


Figure 4.2: The maximum energy voltage recorded at 77.18 volts in maximum time of 4900 seconds recorded through simulation using SDR based radiometer.

The concepts of modulation scheme and order, bandwidth and signal power (re- ferred to hereafter as simply "power") are important to the discussion that follows and thus each warrants a brief introduction here. Power is perhaps the easiest of the three to understand and corresponds to the amount of energy with which a signal is transmitted. The power with which a signal is transmitted affects how far it may propagate and still be correctly received. As the signal propagates through the atmo- sphere, it is attenuated (loses energy) by interactions with molecules in the air. Thus, as the transmitter and receiver move farther apart, all other things being equal, the signal must be transmitted with more power in order to be received.

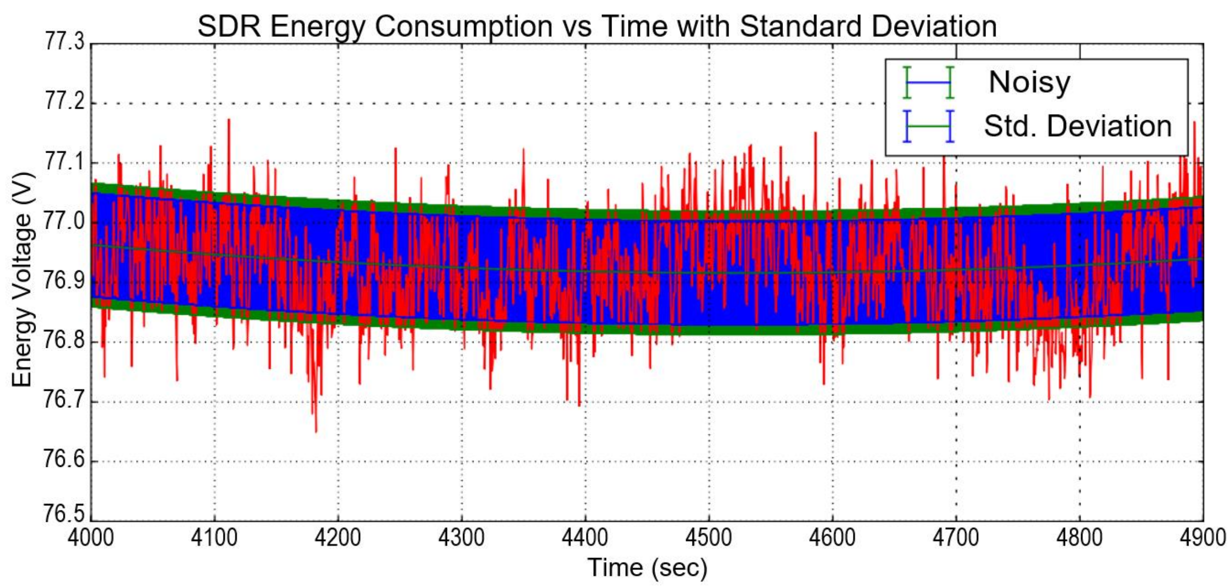


Figure 4.3: The maximum energy voltage recorded at 77.18 volts in maximum time of 4900 seconds recorded through simulation using SDR based radiometer with noisy signals as well as standard deviation.

To investigate each of the independent variables outlined previously and their effect on the network's latency performance, the network is first run without the cognitive engine to collect baseline latency statistics. The engine is then added to the network and the simulation run ten times, once for each value of latency weight and number of generations used to develop solutions. Each simulation is run for 2,000 seconds, with the noise floor starting at -90 dBm and incrementing by 10 dBm every 200 seconds. As the engine runs every 5 seconds, this ensures it runs 40 times for each combination of latency weight or number of generations and noise floor to provide adequate data.

In order to evaluate how the number of generations used to develop solutions affects the engine's runtime, additional tests wherein the engine is removed from the network and run in isolation are also performed. A simple test program that invokes the engine and times its execution is written. The program executes the engine 40 times for each number of generations value and records the engine's runtime for each execution. In investigating how the number of generations affects network performance, the latency data resulting from the five number of generations runs must first be split into ten sets based on the noise floor. For each noise floor, then, there are five data sets, one for each setting of the number of generations parameter, giving network latencies against simulation time. A simple test is performed to compare the average latency without the SDR in the network to that achieved with the engine in the network for each investigated value for the number of generations. A value less than or equal to .05 from this test indicates the engine has an appreciable effect on the network's performance.

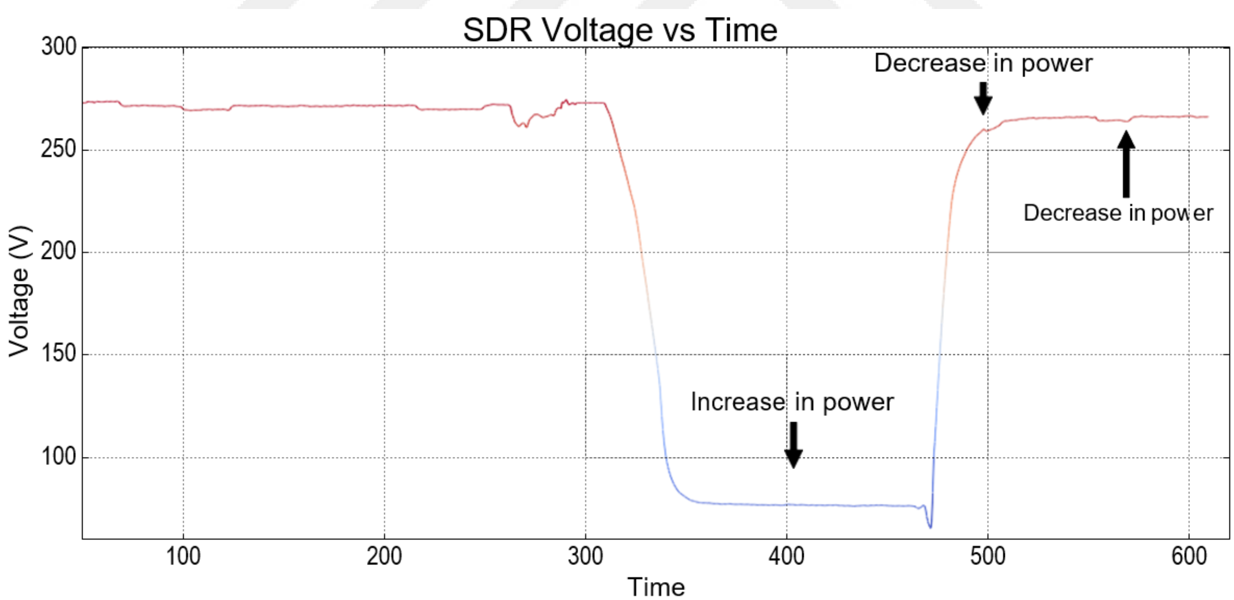


Figure 4.4: The voltage increased recorded at 276.88 volts however voltage reduction recorded at 17.69 volts in constraint to see the positive effect of SDR based radiometer.

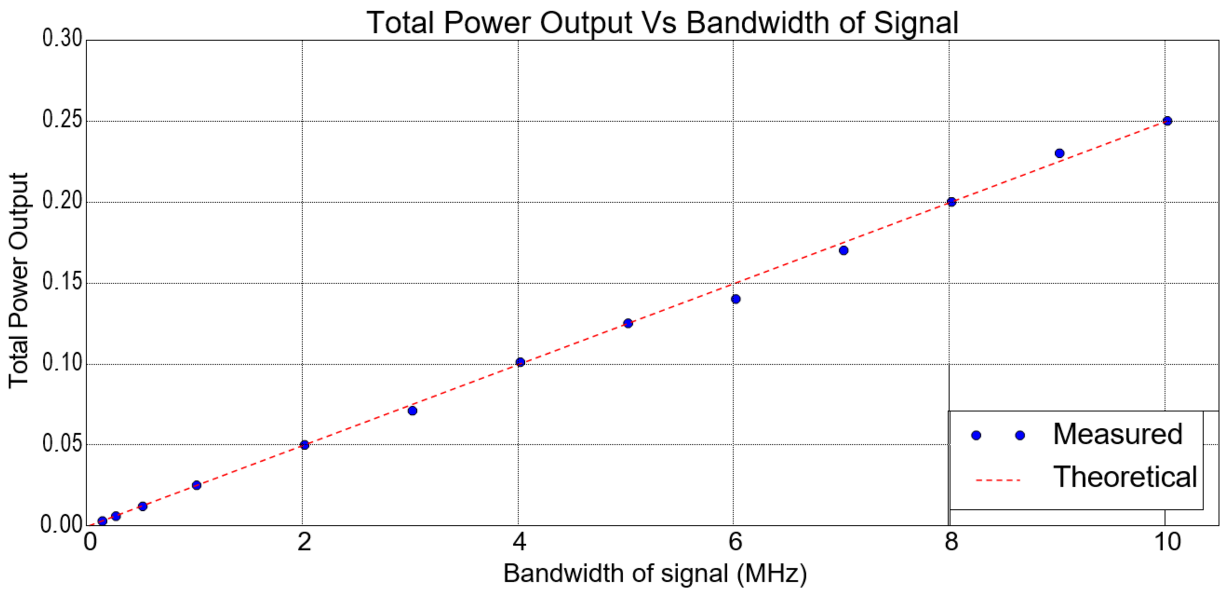


Figure 4.5: The depiction of total power output at 0.25W with respect to bandwidth of signals processed for measured and theoretical outputs with 10 MHz.

The effect latency weight has on network performance is evaluated in a similar manner. The data resulting from the five simulations across which latency weight was altered is first split into ten data sets based on noise floor. t-tests comparing the average latency without an engine in the network to the network's performance with each of the investigated latency weights are performed. As before, a value less than or equal to .05 from these tests indicate the engine with the associated latency weight has an appreciable effect on the network's performance.

By comparison, the data analysis performed on the engine's runtime data is straightforward. As it is expected the runtime will increase in proportion to the number of generations, a linear model is fit to the runtime vice number of generations data. Then, the p-value associated with the number of generations variable is assessed to determine if it statistically significant within the model (i.e., if its value is less than or equal to .05). If it is statistically significant, this indicates the number of generations the engine uses to develop solutions has a direct and substantial impact on its runtime. Additionally, the resulting linear model may be used to determine, given a desired runtime, approximately how many generations may be developed by the engine in that time.

5 DISCUSSION

5.1 DISCUSSION

Since World War II, the human beings have begun to use radio communication sets. It has evolved through various developments in different frequency of operation and different modes of operations. Earlier devices were developed for their own countries utilisation. With global competition and industrialization, the radio sets have crossed across the border. In the last decade or so, every military has started interacting with all friendly foreign countries for conduct of exercises to share the challenges and solutions there of globally. While conducting such exercises with military equipments, Communication equipments such as Radio stand aloof with incompatibility and Inter-operability problem between countries. In Radio Communication, countries use different Frequency ranges from HF(3 to 30 Mega Hertz) to UHF(300 MHz to 3 Giga Hertz) and different modulation techniques for voice and data transmission. In order to solve these challenges, an Unified Communication Architecture is required. Since anything and everything is Software definable in this world and due to rich Mathematics background in Communication field, it is desirable to explore and follow an International Standard in Communication called SCA. Radios which can be implemented using Software are called SDR. In this thesis, the SDR Technology and the management and security of their electronic platforms, the management aspects and few development Tools are discussed.

5.2 SDR TRANSMITTER AND RECIEVER ROLE

In SDR Transmitter the information is formatted into Digital input, mi, in the form of message symbols. Then these symbols could be converted into bit stream, uj, by source encoding, encrypting(for secrecy), channel encoding and multiplexing(for multi signals) process. Depending upon the source coding and multiplexing, the data rate of streams can be controlled. After that desired Pulse coding could be undertaken using modulation such as PAM, QAM, PSK, FSK, OFDM or its variants to form Electromagnetic waves suitable to antenna characteristics. After it travels through media, the signal is received at the receiver.

In SDR the receiver, the received signal from Antenna is demodulated after carrier synchronization and automatic Gain control, using appropriate Demodulation techniques such as OFDM, QAM, FSK, PSK and its variants, to make the bit stream. From Bit Stream, after symbol synchronization and frame synchronization, the message is reconstructed successfully.

5.3 BENEFITS OF SECURITY MANAGEMENT IN SDR

In order to detect BERs that result in reasonably high (but not perfect) latency fitness, the number of bits for which transmission is simulated must be on the order of one million bits. As the inputs to the (possible) IoT encoder, modulator, SDR channel, demodulator and (possible) IoT decoder change for each generation the engine produces, the simulated bit transmissions must be performed each time the engine forms a new solution in order to assign a fitness value to it. Unfortunately, the time required to perform the simulated bit transmission so often causes the MATLAB runtime to greatly exceed that of the MATLAB simulations. Additionally, the SDR network yielded actual latency information whereas the engine reimplementation in MATLAB only yields latency and total fitness information. As a result, the experiments performed and data gathered from the MATLAB engine reimplementation have a slightly different focus than those planned for the MATLAB simulations. The first variable to be assessed through experimentation is IoT encoding. In particular, rather than evaluating multiple schemes to determine the one with which the engine performs best, here the item of interest is whether or not the engine performs better with or without IoT encoding. It is expected IoT encoding will cause an increase in the engine's performance, evidenced by the engine returning solutions with lower BERs and higher latency fitness than those returned for the same non-IoT encoded case. However, it is also expected IoT encoding will increase the runtime as compared to the non-IoT encoded case. Further, this effect is expected to be more pronounced as the noise floor becomes closer to the transmit power (i.e., as the Signal-to-Noise Ratio (SNR) drops). If this trend is indeed exhibited by the data, future work will need to be conducted to determine at what point IoT encoding offers enough of a performance improvement that the penalty to runtime is worth the improved fitness values. Similar to the simulations, another independent variable of interest here is the number of generations used by the engine and its effect on the quality of the solutions the engine returns. As discussed above, due to the manner in which the MATLAB engine reimplementation estimates BER, the number of generations is expected to have a significant impact on the engine's runtime. If this is indeed the case, as with the previous experiments, determining the point at which the increase in runtime does not yield enough improvement in the fitness of returned solutions to warrant using additional generations in finding them becomes an important part of future work.

Table 5.1: Coefficient and SDR value for linear models for fitness of returned solutions and runtime growth of MATLAB engine using large numbers of generations.

SDR Fitness Linear Model			
Noise Floor (dBm)	Term	Value	Voltage value
−50	Linear Term Coefficient	∼ 8.979 ∗ 10−8	∼ .896
	Intercept	∼ .967	∼ 6.393 ∗ 10−208
−30	Linear Term Coefficient	∼ 4.267 ∗ 10−7	∼ .541
	Intercept	∼ .966	∼ 2.252 ∗ 10−207
10	Linear Term Coefficient	∼ −3.853 ∗ 10−7	∼ .731
	Intercept	∼ .955	∼ 8.194 ∗ 10−189
SDR Runtime Linear Model			
Noise Floor (dBm)	Term	Value	Voltage value
−50	Linear Term Coefficient	∼ 1.002	∼ 8.204 ∗ 10−120
	Intercept	∼ 12.365	∼ .000359
−30	Linear Term Coefficient	∼ 1.011	∼ 1.212 ∗ 10−133
	Intercept	∼ .0891	∼ .970
10	Linear Term Coefficient	∼ 1.026	∼ 1.749 ∗ 10−126
	Intercept	∼ −1.648	∼ .566

Unlike SDR modulation scheme, modulation order has a signifi- cant impact on solution fitness. For a noise floor of −50 dBm, the interference is low enough a constellation with 256 points could be used and still maintain high fitness. However, as the noise floor rises to even

−30 dBm, the maximum size the constella- tion may have and still facilitate reliable communication becomes constrained. Using IoT encoding with this noise floor allows a constellation with 128 symbols to be used whereas without IoT encoding, the constellation size is constrained to 64 symbols. As expected, then, IoT encoding allows the receiver to detect and correct bit errors, allowing for a larger constellation size and thus greater throughput. However, this comes at the cost of additional processing and thus greater power consumption. The situation is even more pronounced for a noise floor of 10 dBm. In that case, the interference is so large that a constellation size greater than 4 symbols, even with FEC encoding, results in the receiver not being able to correctly interpret the trans- mitted data.

6 CONCLUSION

6.1 CONCLUSION

In this novel research work, the SDR power consumption is measured using the IoT based genetic algorithm, fitness performance exhibited by the SDR engine may be an indication of its efficiency in finding highly fit solutions. However, a more likely scenario is that the fitness function assigns a disproportionately large number of solutions high fitness values. As a result, work needs to be done to assess the distribution of fitness scores across the search space. Additionally, the likelihood with which the fitness function assigns a high fitness value to a solution that leads to poor network latency performance needs to be assessed. It may be a slight increase in this likelihood is acceptable if it leads to a greater increase in the likelihood of the engine finding a highly fit solution that yields low network latencies. Conversely, it may be that the possibility of the SDR returning a solution that leads to poor performance is acceptable where the maximum voltage recorded at 77.18 volts and thus this probability must be minimized in 4900 seconds. An SDR network that implements IoT based genetic algorithm from which actual latency information may be obtain will enable this work. These experiments should focus on determining whether or not the high probability of the engine finding a highly fit solution as with the experiments presented here carries over to high noise floors with final power output at 0.25W. If not, increased interference may the engine to begin discerning between truly fit solutions and those that would likely not lead to acceptable performance in real-world networks in which the voltage reduction measured at 17.69 volts during experiment.

6.2 FUTURE RECOMMENDATION

In future, it is somewhat unlikely the SDR transmission rate will change; it is set by the user in the scenario configuration file and none of the SDR documentation indicates the nodes may change any of their operational parameters of their own volition. Further, these experiments did not include a cognitive engine that was trying to optimize the nodes’ performance so it does not appear an external entity commanded the transmitter to alter its transmission rate. As for the rate at which the transmitter prepared packets for transmission, if the transmitter did change that rate, it appears to have done so of its own accord. If the transmitter did make the decision to change its operating parameters on its own, it would make measuring latency more difficult.

Analogous queueing within the receiver could also explain the initial linear growth in the latencies. If the receiver has a queue in which it places incoming messages awaiting processing and the rate at which it receives messages is faster than the rate at which it can process them, the amount of time messages wait in the queue to be processed would grow linearly according to the difference in the rate at which they arrive and are processed. This is at least possible; the computation required for processing the received signal to extract the message is considerably more than that required to prepare the message for transmission. As a result, while the transmitter can likely prepare messages faster than it can transmit them, the receiver likely cannot process the incoming signal to recover the communicated information as fast as it arrives. This, however, does not explain the varying growth rate in the two cases. Additionally, the same problem arises here as with the transmitter in explaining the shift to a constant latency. If processing with the receiver explains the data trends, it is unclear why the receiver should have altered its message processing rates or why the message arrival rate should have changed.

REFERENCES

[1]. Cisco Systems Inc. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022 White Paper; Cisco Systems Inc.: San Jose, CA, USA, 2019.

[2]. Wunsch, F.; Paisana, F.; Rajendran, S.; Selim, A.; Alvarez, P.; Muller, S.; Koslowski, S.; Van den Bergh, B.; Pollin, S. DySPAN Spectrum Challenge: Situational Awareness and Opportunistic Spectrum Access Benchmarked. IEEE Trans. Cogn. Commun. Netw. 2017, 3, 550–562.

[3]. Joshi, G.P.; Nam, S.Y.; Kim, S.W. Cognitive Radio Wireless Sensor Networks: Applications, Challenges and Research Trends. Sensors 2013, 13, 11196–11228.

[4]. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. IEEE Internet Things J. 2016, 3, 637–646.

[5]. Ravì, D.; Wong, C.; Lo, B.; Yang, G. A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices. IEEE J. Biomed. Heal. Inform. 2017, 21, 56–64.

[6]. Premsankar, G.; Di Francesco, M.; Taleb, T. Edge Computing for the Internet of Things: A Case Study. IEEE Internet Things J. 2018, 5, 1275–1284.

[7]. Chiang, M.; Zhang, T. Fog and IoT: An Overview of Research Opportunities. IEEE Internet Things J. 2016, 3, 854–864.

[8]. Szilvási, S. Advanced RF Techniques for Wireless Sensor Networks: The Software-Defined Radio Approach. Ph.D. Thesis, Institute for Software Integrated Systems (ISIS), Electrical Engineering, Vanderbilt University, Nashville, TN, USA, 2014.

[9]. Dutta, P.; Kuo, Y.-S.; Lédeczi, A.; Schmid, T.; Völgyesi, P. Putting the Software Radio on a low-calorie diet. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HOTNETS), Monterey, CA, USA, 20–21 October 2010; pp. 1–6.

[10]. Venieris, S.I.; Kouris, A.; Bouganis, C.-S. Toolflows for Mapping Convolutional Neural Networks on FPGAs: A Survey and Future Directions. ACM Comput. Surv. 2018, 51, 1–39.

[11]. Tang, Z.-L.; Li, S.-M.; Yu, L.-J. Implementation of Deep Learning-based Automatic Modulation Classifier on FPGA SDR Platform. Electronics 2018, 7, 122.

[12]. Zhou, A.; Santacruz, S.R.; Johnson, B.C.; Alexandrov, G.; Moin, A.; Burghardt, F.L.; Rabaey, J.M.; Carmena, J.M.; Muller, R. A wireless and artefact-free 128-channel

neuromodulation device for closed-loop stimulation and recording in non-human primates. Nat. Biomed. Eng. 2019, 3, 15–26.

[13]. Xia, L.; Soltan, A.; Luo, J.; Chester, G.; Degenaar, P. A Flash-FPGA based Rodent Control System for Closed-loop Optogenetic Control of Epilepsy. In Proceedings of the IEEE International Symposium on Circuits and Systems, Florence, Italy, 27–30 May 2018; pp. 1–5.

[14]. Polastre, J.; Szewczyk, R.; Culler, D. Telos: Enabling ultra-low power wireless research. In Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), Los Angeles, CA, USA, 15 April 2005; pp. 364–369.

[15]. CrossBow MICAz Datasheet. Available online: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

[16]. Rodriguez-Zurrunero, R.; Utrilla, R.; Romero, E.; Araujo, A. An Adaptive Scheduler for Real-Time Operating Systems to Extend WSN Nodes Lifetime. Wirel. Commun. Mob. Comput. 2018, 2018, 1–10.

[17]. Rodriguez-Zurrunero, R.; Tirado-Andres, F.; Araujo, A. YetiOS: An adaptive operating system for wireless sensor networks. In Proceedings of the 43rd Annual IEEE Conference on Local Computer Networks Workshops (LCN Workshops), Chicago, IL, USA, 1–4 October 2018; pp. 16–22.

[18]. Ettus Research. Available online: https://www.ettus.com/

[19]. Blade RF x40. Available online: https://www.nuand.com/product/bladerf-x40/

[20]. HackRF. Available online: https://greatscottgadgets.com/hackrf/

[21]. WARP Project. Available online: https://warpproject.org/trac/wiki/about

[22]. ADALM-PLUTO. Available online: https://www.analog.com/en/design-center/evaluation-hardware-andsoftware/evaluation-boards-kits/adalm-pluto.html#.

[23]. LimeSDR Boards. Available online: https://limemicro.com/products/boards/

[24]. Rajendran, S.; Lenders, V.; Calvo-Palomino, R.; Cordobes, H.; Pollin, S.; Fuchs, M.; Giustiniano, D.; Van den Bergh, B. Electrosense: Open and Big Spectrum Data. IEEE Commun. Mag. 2017, 56, 210–217.

[25]. Narayanan, R.; Kumar, S. Revisiting Software Defined Radios in the IoT Era. In Proceedings of the 17th ACM Workshop on Hot Topics in Networks, Redmond, WA, USA, 15–16 November 2018; pp. 43–49.

[26]. Tomar, V.S.; Bhatia, V. Low Cost and Power Software Defined Radio Using Raspberry Pi for Disaster Effected Regions. Procedia Comput. Sci. 2015, 58, 401–407.

[27]. RTL-SDR. Available online: https://www.rtl-sdr.com/

[28]. GNU Radio. Available online: https://www.gnuradio.org/

[29]. Szilvási, S.; Babják, B.; Völgyesi, P.; Lédeczi, Á. MarmotE SDR: Experimental platform for low-power wireless protocol stack research. J. Sens. Actuator Netw. 2013, 2, 631–652.

[30]. Kuo, Y.-S.; Pannuto, P.; Schmid, T.; Dutta, P. Reconfiguring the software radio to improve power, price, and portability. In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys), Toronto, ON, Canada, 6–9 November 2012; pp. 267–280.

[31]. Utrilla, R.; Rozas, A.; Blesa, J.; Araujo, A. A Hybrid Approach to Enhance Cognitive Wireless Sensor Networks with Energy-Efficient Software-Defined Radio Capabilities. In Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, Uppsala, Sweden, 20–22 February 2017; pp. 294–299.

[32]. Rodriguez-Zurrunero, R.; Utrilla, R.; Rozas, A.; Araujo, A. Process Management in IoT Operating Systems: Cross-Influence between Processing and Communication Tasks in End-Devices. Sensors 2019, 19, 805.

[33]. Chen, T.; Matinmikko, M.; Chen, X.; Zhou, X.; Ahokangas, P. Software defined mobile networks: Concept, survey, and research directions. IEEE Commun. Mag. 2015, 53, 126–133.

[34]. Foukas, X.; Nikaein, N.; Kassem, M.M.; Marina, M.K.; Kontovasilis, K. FlexRAN: A flexible and programmable platform for software-defined radio access networks. In Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies, Irvine, CA, USA, 12–15 December 2016; pp. 427–441.

[35]. Mijumbi, R.; Serrat, J.; Gorricho, J.L.; Latre, S.; Charalambides, M.; Lopez, D. Management and orchestration challenges in network functions virtualization. IEEE Commun. Mag. 2016, 54, 98–105.

[36]. Karyotis, V.; Stai, E.; Papavassiliou, S. A Component-Based Cross-Layer Framework for Software Defined Wireless Networks. In Proceedings of the 2016 8th IEEE IFIP International

Conference New Technologies, Mobility and Security (NTMS), Larnaca, Cyprus, 21–23 November 2016; pp. 1–6.

[37]. Akhtar, A.M.; Wang, X.; Hanzo, L. Synergistic spectrum sharing in 5G HetNets: A harmonized SDN-enabled approach. IEEE Commun. Mag. 2016, 54, 40–47.

[38]. Autonomic Computing: IBM’s Perspective on the State of Information Technology (manifesto); IBM Research Headquarters: Yorktown Heights, NY, USA, 2017.

[39]. Asghar, M.Z.; Nieminen, P.; Hämäläinen, S.; Ristaniemi, T.; Imran, M.A.; Hämäläinen, T. Towards proactive context-aware self-healing for 5G networks. Comput. Netw. 2017, 128, 5–13.

[40]. Strassner, J.; Raymer, D.; Samudrala, S. Providing seamless mobility using the FOCALE autonomic architecture. In Proceedings of the Next Generation Teletraffic and Wired/Wireless Advanced Networking, NEW2AN 2017, Lecture Notes in Computer Science, St. Petersburg, Russia, 10–14 September 2017; Volume 4712, pp. 330–341.

[41]. Neves, P.; Calé; R; Costa, M.; Gaspar, G.; Alcaraz-Calero, J.; Wang, Q.; Nightingale, J.; Bernini, G.; Carrozzo, G.; Valdivieso, A.; et al. Future mode of operations for 5G–The SELFNET approach enabled by SDN/NFV. Comput. Stand. Interfaces 2017, 54, 229–246.

[42]. Bari, M.F.; Chowdhury, S.R.; Ahmed, R.; Boutaba, R. PolicyCop: An autonomic QoS policy enforcement framework for software defined networks. In Proceedings of the Future Networks and Services (SDN4FNS), Trento, Italy, 11–13 November 2016; pp. 1–7.

[43]. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Turner, J. OpenFlow: Enabling innovation in campus networks. ACM Comput. Commun. Rev. 2018, 38, 69–74.