Performance Investigation of a Modelled Finite-source Cognitive Radio Network

Thesis for the Degree of Doctor of Philosophy (PhD)

by Mohamed Hedi Zaghouani

Supervisor:
Dr. János Sztrik

UNIVERSITY OF DEBRECEN
Doctoral Council of Natural Sciences and Information Technology
Doctoral School of Informatics
Debrecen, 2024

Hereby I declare that I prepared this thesis within the Doctoral Council of Natural Sciences and Information Technology, Doctoral School of Informatics, University of Debrecen in order to obtain a PhD Degree in Informatics at Debrecen University.

The results published in the thesis are not reported in any other PhD theses.

Debrecen, 2024

....

signature of the candidate

Hereby I confirm that Mohamed Hedi Zaghouani candidate conducted his/her studies with my supervision within the Information Technology Systems and Networks with Industrial Applications Doctoral Program of the Doctoral School of Informatics between 2018 and 2024 The independent studies and research work of the candidate significantly contributed to the results published in the thesis.
I also declare that the results published in the thesis are not reported in any other theses.
I support the acceptance of the thesis.

Debrecen, 2024

....

signature of the supervisor

Performance Investigation of a Modelled Finite-source Cognitive Radio Network

Dissertation submitted in partial fulfillment of the requirements
for the doctoral (PhD) degree in informatics

Written by Mohamed Hedi Zaghouani certified Computer Scientist (MSc)

Prepared in the framework of the Doctoral School of Inforamtics of the University of Debrecen
(Information Technology Systems and Networks with Industrial Applications programme)

Dissertation advisor: Dr. János Sztrik

The official opponents of the dissertation:

Dr. ....

Dr. ....

The evaluation committee:

chairperson: Dr. ....

members: Dr. ....

Dr. ....

Dr. ....

Dr. ....

The date of the dissertation defence: .... 20. . .

Contents

1 Introduction **1**

2 The cognitive radio network **5**

2.1 Introduction 6
2.2 Cognitive radio communication 6
2.2.1 Characteristics of cognitive radio 7
2.2.2 Cognitive radio functions 7
2.3 The architecture of cognitive radio network 8
2.3.1 Network architecture 9
2.3.2 Links in cognitive radio network 11
2.4 Spectrum sensing in cognitive radio networks 13
2.4.1 Interference-based detection 14
2.4.2 Non-cooperative detection (Transmitter detection) . . . 15
2.4.3 Cooperative detection 16
2.5 Conclusion 16

3 Simulation modelling of cognitive radio network **19**

3.1 Random processes review 20
3.1.1 Exponential distribution and related distributions . . . 20
3.1.2 Log-normal 22
3.1.3 Gamma 23
3.1.4 Pareto 23
3.1.5 Poisson-process 24
3.1.6 Poisson-process properties 25
3.2 The basics of simulation modelling 26
3.2.1 Discrete-event simulation 26
3.3 Retrial queueing models in cognitive radio network 29
3.3.1 Literature review 29
3.3.2 System model 32
3.4 Simulation of finite-source cognitive radio network 34
3.4.1 Problem statement 34
3.4.2 Logic and organisation of the program 35
3.4.3 Confidence interval, estimation of the mean and variance 40
3.4.4 The batch mean method 41
3.5 Simulation of non-reliable servers in finite-source cognitive radio network 44
3.6 Simulation methodology 46
3.7 Conclusion 47

4 Performance evaluation and analysis **49**

4.1 Introduction 50

4.2 Unreliability Analysis of finite-source cognitive radio 50

4.2.1 System model 51

4.2.2 Validation of results 52

4.2.3 Numerical results 53

4.3 Finite-source cognitive radio network with unreliability and abandonment 67

4.3.1 System model 68

4.3.2 Validation of results 70

4.3.3 Numerical results 70

4.4 Analysis of finite-source cognitive radio network with balking and reneging 79

4.4.1 System model 80

4.4.2 Validation of the results 81

4.4.3 Numerical results 81

5 Conclusion **89**

6 Acknowledgements **93**

Bibliography **95**

1

Introduction

The first chapter describes the retrial queueing model of a cognitive radio network and underlines the motivation of the research conducted in the dissertation.

It is difficult to picture living without networks in our world today as they have become so ingrained in our daily life, especially the internet. We live in a world which is significantly altered by technology, which improves daily. Networks are not excluded from this fast growth, however, however, as is the case in all fields, selecting precise and appropriate frequency bands has proven to be difficult and problematic in wireless networks. Radio spectrums are fundamentally divided up among different users, services, and applications, however, research shows that spectrums are often used relatively infrequently.

The tremendous expansion of wireless communications is causing a reduction in the amount of accessible frequency spectrum, resulting in more congestion in shared radio spectrums. On the other hand, a large number of the pre-allocated frequency bands are underutilized. Static spectrum allocation has always resulted in an under-utilization of the reserved spectrum and wasted resources. Unbending spectrum management is the proven strategy. Every wireless operator is given a unique license to function in a specific frequency band. Finding available bands is difficult when providing new services or enhancing existing ones.

A better spectrum use that helps to overcome this under-utilization issue consists of allowing Dynamic Spectrum Access (DSA). "Cognitive Radio System" is one of the solutions that are based on the DSA. The capability of spectrum sensing, being aware of the working states within the system and then adapting its functioning parameters are the main functionalities of the Cognitive Radio Network (CRN). This strategy seems to be promising to effectively use the frequency band's available spectrum. The term "cognitive radio" refers to a re-configurable radio system that can "cognitively" be adjusted to its user's communications needs, the radio frequency environment in which it works, to other networks and regulatory norms that may be relevant to it.

The cognitive radio reacts to environmental factors by analyzing and using this data for further evaluation. In this approach, users are categorized into two main categories, Primary Users (PU) are authorized users who are granted the exclusive usage of certain licensed frequency bands and Secondary Users (SU). Only when do not conflict with PU, SUs are permitted access to the licensed frequency spectrum. Therefore, two key elements are required for the success of cognitive radios: the ability to recognize an unused spectrum and to temporarily use the spectrum without interfering with primary users.

An overview of cognitive radio systems' architecture is presented in this thesis. The aim is to use queueing theory to create a simulation tool for modelling a cognitive radio network. The efficiency of cognitive radio technologies in serving a specific population is then examined.

The influential work of Erlang in the setting of early telephone exchange networks is where queueing theory first occurred. Since then, it has been used in myriad fields and situations, including industrial engineering, operations research, telecommunications, and data networks. Nowadays, queueing theory got even more popular by dint of cognitive radio networks. Two non-independent frequency bands are needed by the Cognitive Radio Network's Retrial Queueing System to serve the two groups of users: PUs and SUs.

At the licensed spectrum, PUs have preemptive privilege over SUs, whilst SUs are supplied at the unlicensed spectrum with an orbit for the retry users. In various studies, including [2], queueing theoretical methodologies have been applied to represent cognitive radio networks by building up retrial queueing systems with two finite-sources of PUs and SUs, respectively. Since it was assumed that all inter-event times were exponentially distributed, a multidimensional Markov chain was developed. Consequently, the mean and variance of the key performance measures of our system were obtained and presented.

The same system is addressed in this thesis as a natural generalization of the model employed in [2] by implementing non-exponentially distributions for several inter-event times, such: requests generation, service, failure, and repair times.

A stochastic simulation technique is utilized to estimate the most crucial aspects of the system performance, and various case studies highlight the effects of these general distributions.

Simulation has shown a significant ability in modelling and analyzing such complex systems among many methodologies that are beneficial to assess various options. Each simulation research includes at least these elements:

- Converting the specification of a system into an explicit abstraction that permits all interactions in logic and mathematics.
- Determining each abstraction's parameters that need numerical input.
- Identifying all performance metrics whose values need to be estimated.
- Using the data currently available to estimate the values of all unknown input parameters.
- Converting a simulation programming language's logical abstraction to executable code.
- Run the code several times, at least once for each set of input-parameter values, to perform a series of sampling experiments.

- Analyze how well each performance measure's time average resembles its unidentified long-term average.
- Comparing across experiments the relevant sample time averages for each performance metric.

We can estimate the variances, which is one of the benefits of the simulation. The analysis of the sojourn time distribution is often extremely hard to get, and in most instances, its Laplace transform is provided. Although variance estimation is not an easy task, it may be assessed with the use of numerical and algorithmic methodologies. In order to simulate the finite-source cognitive radio network and examine the impact of the inter-event time distributions on the utilization of the main and secondary servers, we developed a simulation program. Some of the study cases we discuss have been combined:

- Finite-source cognitive radio network with non-reliable servers.
- Finite-source cognitive radio network with impatient customers.
- Analysis of CRN with Unreliability and Abandonment combined.
- Analysis of Cognitive Radio Networks with Balking and Reneging.
- Reliability analysis with balking and reneging.
- Performance investigation of reverse balking.
- Performance investigation of reverse balking and reneging on Cognitive Radio Network.

This dissertation presents 5 chapters. The foundation of the topic was provided in Chapter 1 along with a justification of the research study's importance. An overview of cognitive radio networks is given in Chapter 2. Before being able to use cognitive radio to build a network, it is crucial to comprehend cognitive radio communication. The network architecture and several spectrum sensing methods are also covered in this chapter. The premise behind our work is described in Chapter 3. Initially, a general review of discrete-event simulation modelling and fundamental random processes is provided. The simulation model used to evaluate our system is then identified, and all performance measures whose values need estimation are presented with the use of flowcharts. In Chapter 4, all the results gathered during the system model's performance evaluation of the different case studies are displayed and discussed. Different figures and examples with comparisons were used to show the analysis indicated above. The dissertation's completed work and research results are finally summarized in Chapter 5.

2

The cognitive radio network

This chapter provides a general overview of cognitive radio networks' paradigms, design and the used spectrum sensing methods

Contents

2.1 Introduction **6**
2.2 Cognitive radio communication **6**
2.2.1 Characteristics of cognitive radio 7
2.2.2 Cognitive radio functions 7
2.3 The architecture of cognitive radio network **8**
2.3.1 Network architecture 9
2.3.2 Links in cognitive radio network 11
2.4 Spectrum sensing in cognitive radio networks **13**
2.4.1 Interference-based detection 14
2.4.2 Non-cooperative detection (Transmitter detection) . . 15
2.4.3 Cooperative detection 16
2.5 Conclusion **16**

2.1 Introduction

Researchers extensive efforts to enhance wireless technologies face limitations due to the intermittent availability of radio spectrums. As these technologies demand a higher proportion of bandwidth, they are becoming increasingly bandwidth-critical. Spectrum-sensing methods and cognitive radio networks, in particular, have garnered significant interest and exhibit tremendous potential as solutions for enabling the deployment of these advanced technologies.

In a paper published in 1999, [41], Mitola and Maguire first used the sentence "cognitive radio," describing it as a smart radio that recognizes the environment in which it operates and correspondingly changes the transmission process. Since then, the sentence has expanded and been used in a variety of contexts.

As confirmed in [57], one of the most important features of CRNs is their ability to transition between radio access technologies and broadcast in various radio spectrum bands when frequency channel slots (portions) become available. CRNs are often referred to as "secondary networks" as they must coexist with licensed main networks, and have the right to use opportunistically their frequency slots without interfering with them; the term "dynamic spectrum access" refers to this method.

Wireless communication has become more diverse with the advent of cognitive radio networks and their dynamic spectrum access. The ability of transmitters to adapt to varying network congestion, frequency band characteristics, service requirements, and interference represents fundamental criteria. Cognitive radio, by avoiding interference with primary users, transforms the concept of dynamic spectrum access, enabling secondary network users to opportunistically utilize the primary channel. The cognitive radio network is an innovative solution that addresses a spectrum of research challenges, including spectrum hand-off, sensing, medium access control, resource allocation, and more.

2.2 Cognitive radio communication

Energy and bandwidth are the two essential resources of wireless communication. Due to slow improvements in wireless networks' channel capacity and service quality, these resources are becoming less and less useful. Researchers are now focusing on new communication and network models so that these resources can be used accurately and methodically.

2.2.1 Characteristics of cognitive radio

The technology of cognitive radio (CR) is extremely important for networking and communications in the future, as it allows the more effective use of all the resources. It contrasts with conventional communication paradigms in which radio equipment cannot set their operational characteristics, such as transmission power, frequency, modulation type, et cetera, independently of alterations in the surrounding radio environment, as described in [17]. The concepts "cognitive capacity" and "reconfigurability" were defined by Simon Haykin in his work [27] published in 2005. These features are related to the adjustment of the operating mode and the way Cognitive Radios (CRs) gather necessary data. Additionally, cognitive capability empowers these Cognitive Radios to learn and acquire knowledge about transmitted waveforms, communication networks, radio frequency spectrums, user requirements, geographical information, locally accessible resources, and services. By collecting environmental data, CR nodes can dynamically adjust their transmission parameters in response to environmental variables, optimizing their performance.

2.2.2 Cognitive radio functions

Finding open spectrum space, choosing the optimum frequency channel, vacating the frequency as soon as a licensed user shows up and coordinating spectrum access with other users, are all steps in the typical functioning cycle of CR, as shown in Figure 2.1. The following processes help promote such a cognitive cycle:

- *Spectrum sensing*: finding a non-used spectrum and sharing it with other users without any interference.
- *Spectrum handoff and management*: using the optimum spectrum currently available to satisfy user communication needs.
- *Spectrum sharing and allocation*: ensuring that coexisting cognitive users have access to fair spectrum scheduling while transitioning to an improved spectrum, and maintaining the need for smooth communication.

As shown in Figure 2.2, CR may locate the section of the frequency band that isn't occupied by primary users during spectrum sensing. However, if the primary users start utilizing their spectrum once again, CR will be able

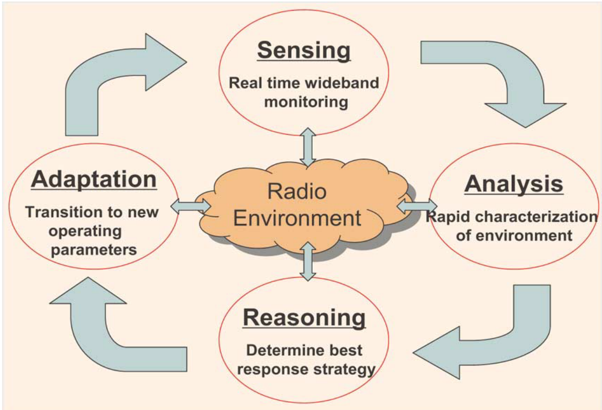


Figure 2.1: Cognitive cycle

to detect this change as well, preventing any negative interference from being brought on by the transmission of the secondary.

The handoff function and spectrum management enable cognitive users to choose the best frequency channel and then hop among multiple bands according to channel characteristics to meet specific Quality of Service (QoS) requirements, in case sensing results in a spectrum-free space (white space) discovery. According to the channel capacity as indicated by noise and interference levels, path loss, channel error rate, holding time, etc., the secondary user utilizing the licensed band, for instance, may reroute his transmission to other accessible frequencies if a primary user reclaims his frequency band.

Thanks to dynamic spectrum access, a secondary user (SU) can share spectrum resources with primary users, other secondary users, or both. To achieve high spectrum efficiency, a suitable method for sharing and allocating spectrum is needed. When secondary users share a licensed band with primary users, the interference caused by secondary spectrum use should be limited to a certain level since primary users have the rights to the spectrum. To reduce collisions and interference, secondary users that share a frequency band should coordinate their access.

2.3 The architecture of cognitive radio network

In order to perform network operations beyond the usage of spectrum white space at the link level, as we explained above, cognitive radio is intended to recognize surrounding networks and communication systems.

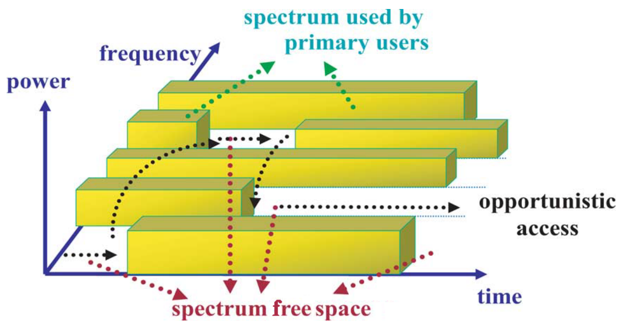


Figure 2.2: Representation of the spectrum's white space

As mentioned in [57], a cognitive radio can be made up of various multi-radio communication systems coexisting, including CR systems. As a heterogeneous network, the CRN consists of several communication technologies. User terminals, wireless access technologies, service providers, apps, networks, etc., all exhibit heterogeneity. [1].

2.3.1 Network architecture

Primary and secondary networks are two subsystems that make up the cognitive network architecture, as seen in Figure 2.3. The following definitions describe the key components of the two groups of the system:

2.3.1.1 Primary network

Such as in TV broadcasting and common cellular, the primary network is an existing infrastructure profiting from granted licensing rights for a certain frequency band. The essential components of the primary network are:

- *Primary user (PU)*: Only a licensed user is permitted to use a certain frequency band. The primary base station's capacity to control this access shouldn't be interfered with by the actions of other unlicensed (secondary) users. Licensed users don't need any modifications or additional features in order to coexist with secondary base stations and cognitive users.

- *Primary Channel Service (PCS)*: A fixed infrastructure network component with a spectrum license, such as the base-station transceiver system (BTS) in a cellular system, is known as a primary base station (licensed

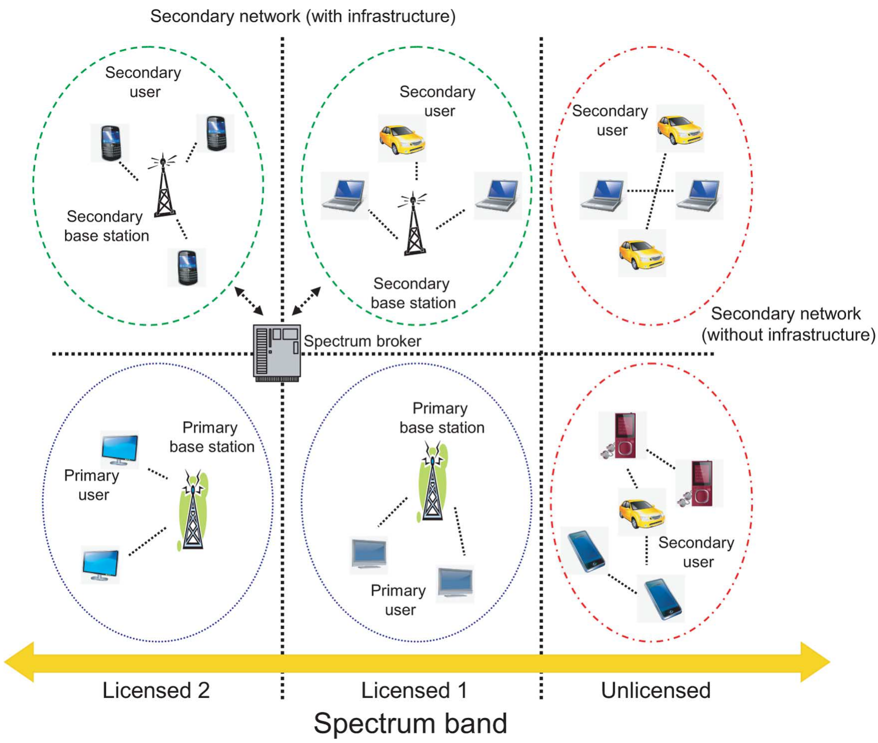


Figure 2.3: The architecture of cognitive radio network

base station). The principal base station does not, in theory, have the cognitive capacity to share a spectrum with secondary users. However, for main network access by secondary users, it could be necessary for the principal base station to include both legacy and cognitive protocols, as discussed below.

2.3.1.2 Secondary network

The secondary network (also known as an unlicensed network, Dynamic Spectrum Access network, or cognitive radio network) is not authorized to perform in the licensed band, therefore, only opportunistic access to the spectrum is permitted. As illustrated in Figure 2.4, cognitive networks may be implemented as both an infrastructure network and an ad-hoc network. A secondary network's main components are as follows:

- *Secondary user (SU)*: often known as a cognitive user, lacks a frequency band license. As a result, more features are needed to share the licensed

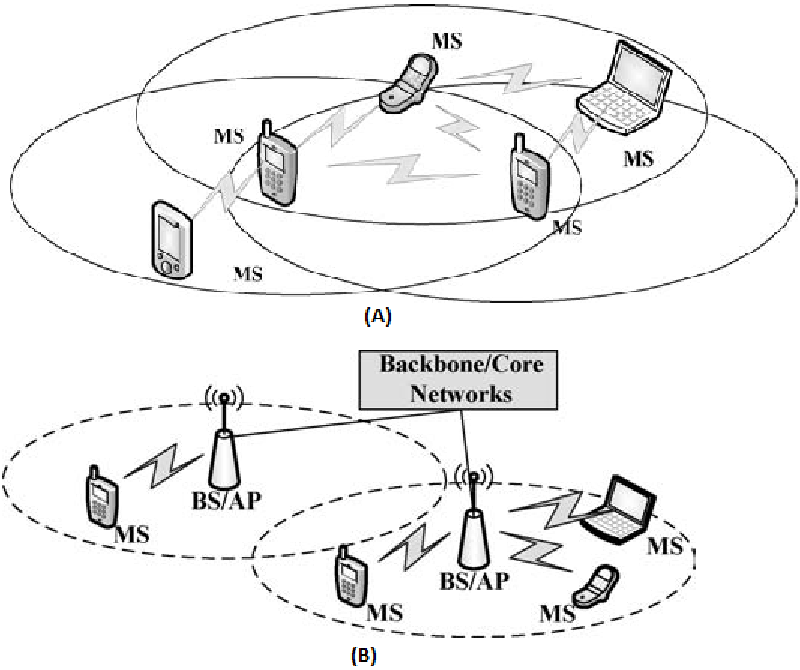


Figure 2.4: (A) Ad-hoc architecture/(B) Infrastructure architecture

spectrum band.

- *Secondary Channel Service (SCS)*: known also as a cognitive (secondary) base-station, is a permanent element of the infrastructure. SCS offers secondary customers a single-hop connection with the lack of spectrum access license. An additional user may connect to other networks using this connection.

- *Spectrum broker*: occurs when numerous secondary networks use a single frequency band. In this scenario, a centralized network component known as a spectrum broker may coordinate spectrum utilization. It is possible to connect to each network and act as a spectrum information manager to allow several cognitive networks, [9, 32, 55].

2.3.2 Links in cognitive radio network

CRNs may serve as a reminder of two different wireless communication system types: primary systems (PS) and cognitive radio (CR) systems, which are grouped on frequency bands based on their relative importance. An established

Table 2.1: Summary of links in the CRN

Rx\Tx	SU	SCS	PU	PCS
SU	•	•	•	•
SCS	•	•	•	
PU	•	•	•	•
PCS	•		•	•

system using one or more fixed frequency bands is referred to as a main system. In the same geographic area or the same frequency band, many major system types operate in licensed or unlicensed bands.

A cognitive radio system is not granted access to certain frequency ranges. Entities of the CR system must communicate with one another via dynamic spectrum gaps (holes) and opportunistic access.

Some inter-system connections need to be allowed since the CR system may provide compatibility among various communication systems. Table 3.1 and Figure 2.5 both list and depict the connection's possible configurations.

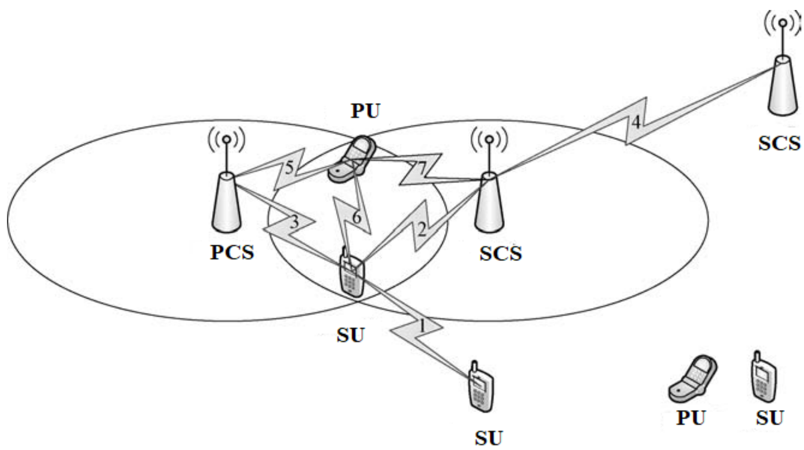


Figure 2.5: Links in CRNs

- **SU ⇌ SU**: SUs may communicate directly with one another. In different frequency bands that may be licensed or unlicensed as their operating frequency band, they can cooperate and detect spectrum use gaps.
- **SU ⇌ SCS**: SCS can acquire additional sensing data from Mobile Stations, dynamically detect a frequency range surrounding it, and provide one-hop access to SU inside its service area. It could be necessary to

use collaborative sensing for this purpose. Under the direction of SCS, the SU may connect to backbone networks or communicate with other communications systems.

- **SU** $\rightleftharpoons$ **PCS**: A SU will change its configuration and merge with the primary system if a connection to a PCS is required. It will then attempt to occupy that band as a new user.
- **SCS** $\rightleftharpoons$ **SCS**: They may build a mesh wireless backbone network while providing direct wireless connections between SCSs. They can dynamically choose an operating frequency band and communicate with one another thanks to their cognitive radio skills.
- **PU** $\rightleftharpoons$ **PCS**: The connection between users and channel services is the standard one-hop connection. The PCS is in charge of organizing communications within its service area and enabling the PU access to the backbone network. This connection is fundamentally different from other linkages, it is always bi-directional.
- **PU** $\rightleftharpoons$ **SU**: This form of connection could be required to offer compatibility between various communication platforms. The SU will then restructure itself into a component of the primary system in this situation.
- **PU** $\rightleftharpoons$ **SCS**: This sort of connection could be required to offer compatibility between various communication platforms. The SCS can only provide the PU access service if it can execute the main system's protocol.
- **PU** $\rightleftharpoons$ **PU**: In wireless networking systems, this method of communication may take place in the main system as a kind of ad-hoc network. However, it could also be disallowed in infrastructure mode in certain systems.

According to Kwang Cheng Cheng and Ramjee Prasad's description of these connectivity opportunities in cognitive radio networks in [13], CR links are unique among the options listed above. Every single one of the other seven kinds of linkages is only accessible in one way during a window of opportunity for spectrum access, except the connection between PUs and PCS, which ensures a bidirectional connection.

2.4 Spectrum sensing in cognitive radio networks

From a design perspective, cognitive radio (CR) users are considered residents in the spectrum they occupy. To achieve this, efficient spectrum management services must identify and occupy free channels without disrupting the services of primary users (PUs) and promptly vacate channels when PU activity is detected. The successful implementation of these concepts relies on the CR user's awareness of their surroundings, facilitated by spectrum sensing systems.

Through spectrum sensing, CR can obtain crucial information about its radio environment, such as the presence of primary users and the availability of spectrum gaps. Spectrum gaps refer to unused portions of the spectrum. One challenge faced by cognitive radios is the accurate measurement of a channel between a main receiver and a transmitter in real-world scenarios.

The information obtained through spectrum sensing allows the CR to adjust its transmission and reception characteristics, including modulation schemes, frequency, and transmission power, to ensure efficient spectrum utilization.

Finding the licensed users who receive data within a secondary user's communication range is the most effective technique to find spectrum gaps. It is challenging for a cognitive radio to directly assess the channel between a primary a transmitter and receiver. As a result, the newest researches focus on the identification of main transmitters based on nearby secondary user observations. According to Figure 2.6, three categories may be used to categorize spectrum sensing techniques: interference-based management, cooperative detection, and transmitter identification.

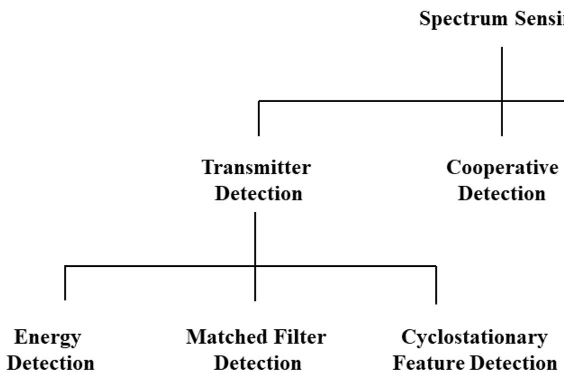


Figure 2.6: The different classes of spectrum sensing techniques

2.4.1 Interference-based detection

The majority of interference regulations focus on the transmitter, making it possible to control interference at the transmitter by adjusting the radiated power, out-of-band emissions, and specific transmitter positions. Interference typically originates at the receivers. As a result, the FCC [17] has unveiled a new interference measurement standard termed interference temperature. In contrast to the conventional transmitter-centred approach, the interference temperature model handles interference at the receiver using the interference temperature limit, where the amount of new interference the receiver can take is the determining factor. In other words, the interference temperature model accounts for the total RF energy from various transmissions and provides the maximum upper limit for the amount of that energy. Cognitive users are allowed to use the spectrum band so long as they do not communicate over it.

2.4.2 Non-cooperative detection (Transmitter detection)

CR users are unable to comprehend the precise details of current transmissions within primary networks since it is often assumed that they have no real-time connection with primary receivers and transmitters. The same users only pick up on the signal from a main transmitter via their local observations to differentiate between utilized and underutilized frequency bands in transmitter detection. To identify whether a signal from the main transmitter is locally present in a certain spectrum, CR users should be able to do so. As seen in [25], the fundamental hypothesis model for transmitter detection may be described as follows:

$$x(t) = \begin{cases} n(t) & H_0 \\ hs(t) + n(t) & H_1 \end{cases} \tag{2.1}$$

where $x(t)$ is the received signal by a CR user, $n(t)$ is a zero-mean additive white Gaussian noise (AWGN), $s(t)$ is the primary user's transmitted signal and h is the amplitude gain of the channel. H_0 is a null hypothesis, according to which there is no signal from a licensed user in a certain spectrum band. The alternative hypothesis H_1, on the other hand, suggests that there is a primary user signal [21]. For transmitter identification, three methods are often employed: feature detection, matched filter detection, and energy detection. As defined and discussed in [52, 46, 54].

- ***Feature detection***: Cyclostationary feature detection is an alternate detection technique. In order to provide a built-in periodicity, modulated signals are often connected with pulse trains, sine wave carriers, hopping sequences, repeated spreads, or cyclic prefixes. The periodicity of the mean value and autocorrelation of these modulated signals is known as cyclostationarity. The examination of a spectral correlation function yields these features.

- ***Matched filter detection***: The matched filter for optimizing the signal-to-noise ratio (SNR) in the presence of additive stochastic noise is the ideal detector in stationary Gaussian noise when the information of the primary user signal is known to the CR user. To meet a detection error probability constraint, this detection approach only needs $O(1/SNR)$ samples.

- ***Energy detection***: The ideal detector is an energy detector if the receiver is only able to determine the power of random Gaussian noise as the principal user signal, for example. The energy of the main signal received by CR users allows them to determine the presence or absence of the primary users using the energy detection technique. The received signal is squared and integrated over the observation period to determine the main signal's energy. Finally, a threshold value is used to determine whether or not a primary user is present by comparing the integrator output with it.

2.4.3 Cooperative detection

Transmitter detection systems rely solely on the weak signals of primary transmitters due to the absence of communication between primary users and Cognitive Radio (CR) users. Without information from the main receiver, transmitter identification algorithms may inadvertently interfere with primary receivers. Moreover, these models struggle to address the hidden terminal issue, as CR transmitters may have a good line of sight for CR receivers, but shadowing might hinder the location of the primary transmitter. Cooperative detection, a process involving the collaboration of data from multiple users, enhances the accuracy of main transmitter detection. By facilitating the identification of a single user, cooperative detection proves to be more successful. Additionally, it can mitigate the effects of multi-path fading and shadowing, thereby increasing detection probability, especially in scenes with significant shading.

2.5 Conclusion

By adopting opportunistic use of the existing available wireless spectrum, cognitive networks are being created to address present wireless network difficulties brought on by the restricted availability and wasteful use of spectrum. An overview of their primary functionality, architectural designs, and spectrum sensing methods has been presented in this chapter. The simulation modelling of a cognitive radio network with a finite number of sources using queuing theory is detailed in depth in the next chapter, where our system model is also shown.

3

Simulation modelling of cognitive radio network

The theoretical and experimental approaches employed in this chapter's scientific study are described

Contents

3.1 Random processes review 20
3.1.1 Exponential distribution and related distributions . . 20
3.1.2 Log-normal 22
3.1.3 Gamma 23
3.1.4 Pareto 23
3.1.5 Poisson-process 24
3.1.6 Poisson-process properties 25
3.2 The basics of simulation modelling 26
3.2.1 Discrete-event simulation 26
3.3 Retrial queueing models in cognitive radio network . 29
3.3.1 Literature review 29
3.3.2 System model 32
3.4 Simulation of finite-source cognitive radio network . 34
3.4.1 Problem statement 34
3.4.2 Logic and organisation of the program 35
3.4.3 Confidence interval, estimation of the mean and variance 40
3.4.4 The batch mean method 41
3.5 Simulation of non-reliable servers in finite-source cognitive radio network 44
3.6 Simulation methodology 46
3.7 Conclusion 47

3.1 Random processes review

This section of the thesis covers the distributions and processes used in the evaluation of our queuing system. We'll discuss the pertinent distributions based on the analytical results.

3.1.1 Exponential distribution and related distributions

The study of simple queueing systems relies on several fundamental ideas and methods from the theory of stochastic processes. The characteristics of the exponential distribution and the Poisson-process are the most significant of them. Many of the studies and findings in the stochastic analysis are built on the supposition that exponential distributions are the intervals between occurrences in the stochastic processes being examined. Therefore, a summary of these distributions' properties will be provided in this section.

The memoryless feature of the exponential distribution, independent of its associated distributions, is defined as follows:

Definition 3.1.1 *A random variable $\bar{x}$ is considered to be memoryless if, and only if, for every $\alpha, \beta \geq 0$,*

$$P\{\bar{x} > \alpha + \beta \mid \bar{x} > \beta\} = P\{\bar{x} > \alpha\} \tag{3.1}$$

For instance, flipping a fair coin is memoryless. We have a 50% probability of getting heads every time we toss the coin. In other words, the exponential distribution is memoryless since its conduct in the past will not affect the future. No matter how much time has passed, every moment is like the start of a new random period with the same distribution.

Definition 3.1.2 *Only if its density function is provided by the following, a random variable X considered to have an exponential distribution with parameter λ where $\lambda > 0$ ($X \in Exp(\lambda)$)*

$$f(x) = \begin{cases} 0, & \text{if } x < 0. \\ \lambda e^{-\lambda x}, & \text{if } x \geq 0. \end{cases} \tag{3.2}$$

its density function is:

$$F(x) = \begin{cases} 0, & \textit{if } x < 0. \\ 1 - e^{-\lambda x}, & \textit{if } x \geq 0. \end{cases} \tag{3.3}$$

Consequently, for a variation X, the variance, the mean and the square coefficient of variation are:

$$E(X) = \frac{1}{\lambda}, \quad Var(X) = \frac{1}{\lambda^2}, \quad C_x^2 = 1. \tag{3.4}$$

3.1.1.1 Hypo-exponential

The hyper and hypo-exponential distributions are special forms of phase-type distributions that are useful in queuing theory. These distributions, which are obtained by splitting the whole period into several phases, each with an exponential distribution whose parameters may be the same or different, serve as models for inter-arrival times or service times in queuing systems.

Definition 3.1.3 *Assuming that $X_i \in Exp(\lambda_i)$ $(i = 1, ..., n)$ is independent exponentially distributed random variables. The random variable $Y_n = X_1 + ... + X_n$ has a hypo-exponential distribution if and only if its density function is:*

$$f_{Y_n}(x) = \begin{cases} 0, & \textit{if } x < 0. \\ (-1)^{n-1}[\prod_{i=0}^{n} \lambda_i] \sum_{j=0}^{n} \frac{e^{-\lambda_j x}}{\prod_{k=1,k\neq j}^{n}(\lambda_j - \lambda_k)}, & \textit{if } x \geq 0. \end{cases} \tag{3.5}$$

Hence, the following are the essential features of the hypo-exponential distribution:

$$E(Y_n) = \sum_{i=1}^{n} \frac{1}{\lambda_i} \quad \text{and} \quad Var(Y_n) = \sum_{i=1}^{n} \frac{1}{\lambda_i^2} \tag{3.6}$$

With a squared coefficient of variation:

$$C_{Y_n}^2 = \frac{\sum_{i=1}^{n}(\frac{1}{\lambda_i})^2}{(\sum_{i=1}^{n}\frac{1}{\lambda_i})^2} \leq 1. \tag{3.7}$$

3.1.1.2 Hyper-exponential

A number of separate exponential distributions come together to form the hyper-exponential distribution. The following is its density function:

Definition 3.1.4 *Let* $X_i \in Exp(\lambda_i)$ $(i = 1, ..., n)$ *and* $p_1, ..., p_n$ *be distributions. When the following formulas describe the density function of the random variable* Y_n*, it is said to have a hypo-exponential distribution:*

$$f_{Y_n}(x) = \begin{cases} 0, & \textit{if } x < 0. \\ \sum_{i=0}^{n} p_i \lambda_i e^{-\lambda_i x}, & \textit{if } x \geq 0. \end{cases} \tag{3.8}$$

It is easy to see that

$$E(Y_n) = \sum_{i=1}^{n} \frac{p_i}{\lambda_i} \quad \text{and} \quad E(Y_n^2) = 2\sum_{i=1}^{n} \frac{p_i}{\lambda_i^2} \tag{3.9}$$

It can be shown that:

$$C_{Y_n}^2 = \frac{\sum_{i=1}^{n} p_i \frac{2}{\lambda_i^2} - (\sum_{i=1}^{n} p_i \frac{1}{\lambda_i})^2}{(\sum_{i=1}^{n} p_i \frac{1}{\lambda_i})^2} \geq 1. \tag{3.10}$$

3.1.2 Log-normal

Let $Y \in N(m, \sigma)$, then the random variable $X = e^Y$ is said to have log-normal distribution with parameters (m, σ), $X \in LN(m, \sigma)$

$$F_x(x) = \phi(\frac{lnx - m}{\sigma}), x > 0 \tag{3.11}$$

$$f_x(x) = \phi'(\frac{lnx - m}{\sigma}) = \frac{1}{\sigma x}\varphi(\frac{lnx - m}{\sigma}), x > 0 \tag{3.12}$$

It can be proved that:

$$E(X) = e^{m+\frac{\sigma^2}{2}} \quad , \quad Var(X) = e^{2m+\sigma^2}(e^{\sigma^2} - 1) \quad \text{and} \quad C_X^2 = e^{\sigma^2} - 1 \tag{3.13}$$

3.1.3 Gamma

A random variable X has a Gamma distribution if its density function is the following, which is a broad form of statistical distribution:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{\beta(\beta x)^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, & \text{if } x \geq 0 \end{cases} \tag{3.14}$$

where $\beta > 0$ and $\alpha > 0$.

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt \tag{3.15}$$

This so-called full gamma function contains two parameters, α and β, which are referred to as the "shape parameter" and the "scaling parameter," respectively. These two variables serve as the random number generator's input parameters. The squared coefficient of variation of the random variable X is known as $C_x^2 = \frac{Var(X)}{(EX)^2}$.

The following may be used to determine the mean value, variance, and squared coefficient of variation:

$$\overline{X} = \frac{\alpha}{\beta}, \qquad Var(X) = \frac{\alpha}{\beta^2}, \qquad C_x^2 = \frac{1}{\alpha} \tag{3.16}$$

The following is used to derive parameters α and β from a determined mean value and variance:

$$\alpha = \frac{1}{C_x^2}, \qquad \beta = \frac{\alpha}{\overline{X}} \tag{3.17}$$

3.1.4 Pareto

If a random variable X has the following density function, it has a Pareto distribution:

$$f(x) = \begin{cases} 0 & \text{if } x < k \\ \alpha k^\alpha x^{-\alpha-1}, & \text{if } x \geq k \end{cases} \tag{3.18}$$

with a distribution function:

$$F(x) = \begin{cases} 0 & \text{if } x < k \\ 1 - (\frac{k}{x})^{\alpha}, & \text{if } x \geq k \end{cases} \tag{3.19}$$

where $\alpha, k > 0$

With two input parameters of random number generations, α (shape parameters) and k (location parameter), the mean value, variation and squared coefficient of variation can be obtained:

$$\overline{X} = \begin{cases} \frac{k\alpha}{\alpha-1} & \text{if } \alpha > 1 \\ \infty & \text{if } \alpha \leq 1 \end{cases}, \qquad Var(X) = \frac{k^2\alpha}{\alpha-2} - (\frac{k\alpha}{\alpha-1})^2 C_x^2 = \frac{(\alpha-1)^2}{\alpha(\alpha-2)} - 1, \qquad \alpha > 2 \tag{3.20}$$

3.1.5 Poisson-process

The Poisson process, a widely employed model in various fields, describes the occurrence of rare events over time. In the context of counting events such as calls received in a call center, births at a hospital, or arrivals at a service facility, the Poisson process is often utilized. The Poisson process is one of the most significant models used in queueing theory, representing the arrival process of events. In the context of teletraffic theory, these 'events' could refer to calls or packets. When calls or packets originate from a large number of different sources or users, the Poisson process is the appropriate model to employ.

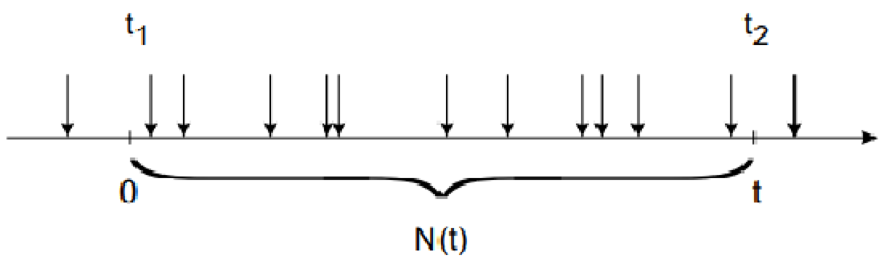


Figure 3.1: number of arrivals in the interval $(0, t)$

The so-called counter process $N(t)$ mathematically describes the process. The counter displays the total number of arrivals that have taken place between $(0, t)$ or, more broadly, (t_1, t_2).

Assuming λ an arrival intensity, the Poisson-process might be referred to as a pure birth process. The inter-arrival times are independent, and Expçλ where $Pinterarrivaltime > t = e^{-\lambda t}$.

3.1.6 Poisson-process properties

- In function to the number of arrivals:

  Given that there are $N(t) = n$ arrivals in the interval $(0, t)$, these n arrivals are evenly and independently distributed across the interval. The following is an example of how to produce a Poisson-process in the range $(0, t)$:

  - Take the $Poisson(\lambda t)$ distribution and subtract the entire number of arrivals, n.
  - draw each arrival's location, independently of the others, from the uniform distribution in the range $(0, t)$.

- Superposition:
  It is considered to be a Poisson-process with intensity $\lambda = \lambda_1 + \lambda_2$, if the superposition of two Poisson-processes have intensities λ_1 and λ_2.

- Random selection
  The process is Poisson with intensity $p\lambda$, if a random selection is performed from a Poisson-process with intensity λ such that each arrival is picked with probability p, independently of the others.

- Random split:
  The resultant processes are separate Poisson-processes with intensities $p1\lambda$ and $p2\lambda$ if a Poisson-process with intensity λ is arbitrarily split into two sub-processes with probability $p1$ and $p2$, where $p1 + p2 = 1$. (This finding enables a simple generalization to a split into more than two sub-processes).

- PASTA:
  The so-called PASTA (Poisson Arrivals See Time Averages) property applies to the Poisson-process (: customers who have Poisson arrivals, for instance, see the system as if they entered it at a random moment in time (despite they induce the evolution of the system).

3.2 The basics of simulation modelling

Simulation and computational modelling have become one of the most powerful design tools in the industry, especially when used to analyze and manage dynamic systems. It is one of the most popular methodologies in operations research and management science, if not the most popular one. This is evidenced by the Winter Simulation Conference, which draws 600 to 800 people annually. Additionally, many other simulation conferences usually have more than 100 attendees each year. For a look at the historical development of simulation modelling, see Nance and Sargent (2002).

3.2.1 Discrete-event simulation

Through a representation in which the state variables change immediately at different time points, discrete-event simulation models a system's development through time. An event is said to occur at these moments in time when it is characterized as an instantaneous phenomenon that may change the system's status.

Discrete-event simulation models have been used to simulate a wide range of real-world systems, but they all share a few common components. For each of these components, there is a logical organization that encourages scripting, testing, and eventual modification of the simulation model program. In particular, most models of discrete-event simulation will have the following elements utilizing the next-event time-advance approach coded in a general-purpose language:

- **System state:** Being a set of state variables, they are required to characterize the system at a certain moment.
- **Simulation clock:** Variable that stores the current value of simulated time.
- **Event list:** A list of the future occurrences of each event's type.
- **Statistical counters:** Variables for keeping track of numerical system performance data.
- **Initialization routine:** A sub-program to start the simulation model at time 0.

- **Timing routine:** The next event is selected from the list by a sub-program, and the simulation clock is then set to the time the event will occur.

- **Event routine:** A sub-program allows the system's state modification if a specific type of event takes place.

- **Library routines:** A set of sub-programs that the simulation model uses to produce random observations from probability distributions.

- **Report Generator:** A separate program that, when the simulation is complete, estimates the necessary performance metrics using the numerical counters and creates a report.

- **Main program:** a sub-program that initiates the timing routine to evaluate the impending event and then transfers control to the right event routine to appropriately update the system state. The main program may check for completion and run the report generator when the simulation is completed.

The flow of the diagram between these elements is shown in Figure 3.2. The simulation starts at time 0 with the main program performing the system state, the initialization routine which initializes the simulation clock, the event list, and the statistical counters.

When the main program has gained control again, it calls the timing routine to figure out the most likely sort of occurrence. If another type i event is expected to happen after this one, the simulation clock is extended until that time, and control is then sent back to the main program.

The main program then launches the event routine i, where typically three different sorts of actions take place: The first involves changing the system state to reflect the occurrence of an event of type i. The second step is obtaining data on system performance by updating the statistical counters, followed by computing the future event's occurrence timings and adding this information to the event list. It is often essential to create random variables (observations) from probability distributions to specify these future event timings or inter-event periods.

A test is often run to determine if the simulation should now be stopped when all processing has been finished, either in the event procedure i or in the main program. If the simulation is to be stopped, the main program will call the report generator to estimate the appropriate performance measures (from the statistics counters) and produces a report. If the simulation is not yet finished, a control cycle is repeated between each component until the simulation's end condition is finally met.

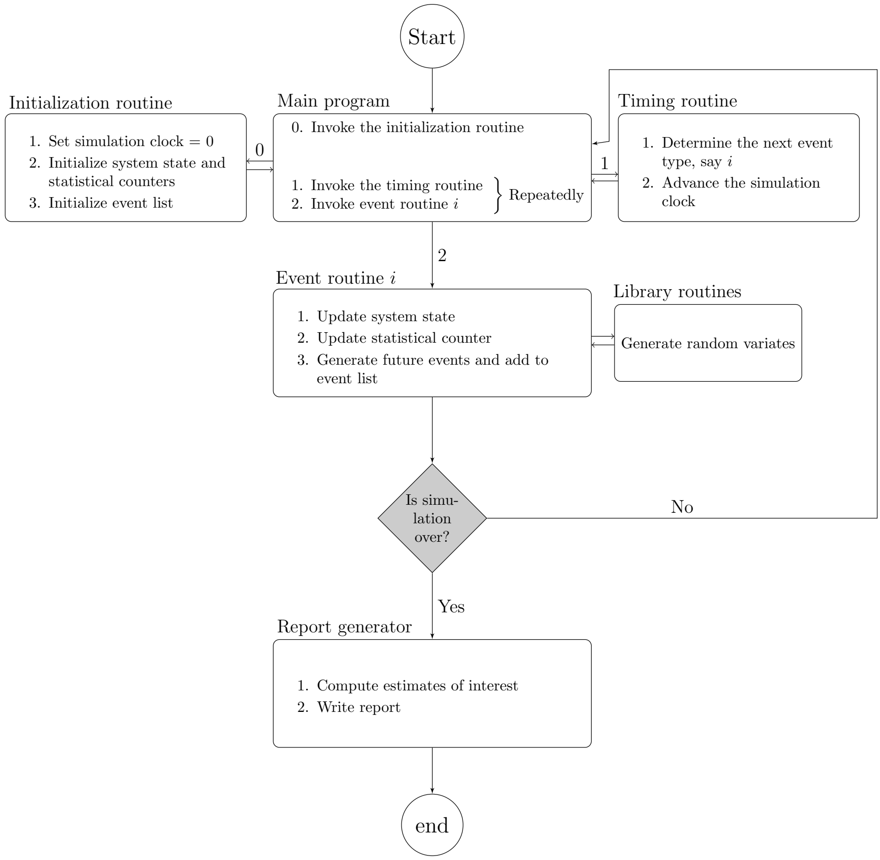


Figure 3.2: The control flow of the next-event time-advance approach

A few further remarks on the system state may be appropriate before moving on to the following section, which will go into depth about the modelling of the finite-source retrial queueing system for cognitive radio networks employing simulation.

We are dealing with a collection of entities that are identified by a set of data values, known as attributes. The system state for a discrete-event simulation model includes these attributes. Each entity has a record in the list of the entity's characteristics, and a certain rule determines the order in which the records are added to the list.

Also, when designing such simulations in a general-purpose programming language like C, the form and functionality of a discrete-event simulation program employing the next-event time-advance approach are relatively usual. This kind of simulation modelling is known as the *event scheduling approach*.

3.3 Retrial queueing models in cognitive radio network

Retrial queues are characterized by the following feature: a job that discovers all servers occupied upon arrival leaves the service facility and retires after a random interval of time. For the performance evaluation of computer and communication systems, queuing models are often utilized. Retrial queues can be used in the performance modelling of many real-world systems. For instance, they can be used to model magnetic disk memory systems [44], cellular mobile networks [50], computer networks [29], local-area networks with non-persistent CSMA/CD protocols [30], and star topologies [34, 40]. Additional recent results using finite-source primary requests may be found in the following citations: [3, 4, 5, 15]. This section introduces an introduction to the finite-source retrial queue that was developed to simulate the cognitive radio network for this thesis.

3.3.1 Literature review

The Cognitive Radio Network (CRN) is often modeled based on fundamental assumptions, with the prioritization of primary users (PUs) over secondary users (SUs) and the unreliability of servers being among the most common. In the CRN paradigm, PUs are granted priority over SUs' transmissions, utilizing servers as channels that SUs can opportunistically and dynamically access under specific conditions. Requests, in the form of data packets, sessions, or connections from both PUs and SUs, are queued when immediate access to the required channel is not feasible.

Researchers have delved into the complexity of models encompassing various interacting factors affecting queueing analysis in Cognitive Radio Networks (CRNs). These factors include PU/SU homogeneity, homogeneous channels, negligible spectrum sensing, perfect spectrum sensing, and channel connection time. The analysis involves extracting performance measures from queued PU and SU data packets, sessions, or connections, emphasizing their inability to immediately access the required channel.

Let's have a look at some particular queueing models from the literature that have used continuous time models in the context of CRNs.

Many performance measures like the efficiency of a system, spectrum occupancy and delay and of two CRN queues (preemptive priority PU M/M/1 queue and retrial SU M/M/1 queue), both served by the same server were assessed by Chang and Jang in [11].

New arriving SU customers finding the spectrum busy, exit the service unit and are routed to the orbit, where they retry to request service again at the SU queue with a specific number of retries. The steady-state probability, the typical number of users in the system, and the typical latency are calculated from the results of [26].

In [16], the authors suggest and examine a priority retrial queue paradigm for CRN. Different PU types are claimed to have several service time distributions. There is only one kind of SUs and one sub-channel dedicated to them. PU requires the whole channel. A marked Markovian arrival process (MMAP) is used to simulate PU and SU arrivals.

While SU service times are exponentially distributed, PU service time distributions are described using phase-type distributions. SUs that find the sub-channel busy move into an orbit from which they retry to access the system (based on a random retrial time, exponentially distributed), or permanently exit the orbit out of impatience, PUs that find the server busy are lost (PUs have preemptive priority over SUs). SUs that are aborted by the arrival of PU, also enter orbit. The number of SUs in orbit, the number of PU customers in service, the number of SU customers in service, the state of the underlying MMAP, and the number of servers at a certain phase for PU customers are all states of the continuous-time Markov chain that is being modelled.

In a system with priority preemption, Heo et al. [28] examine the SU flow performance. A distributor provides the SUs with a number of frequency service bands that are used to settle the system. The blocking and forced termination probability of the SUs were also calculated by the authors.

In their published study [38], Liu et al. evaluated a CRN while taking the PU's traffic pattern into account. The ON-OFF behavior is used and evaluated, it is found interesting even though it is often considered as being insufficient. A generic Gaussian distribution is enforced for the channel capacity with the assumption of fading. The researchers study the system performance under fractional Brownian motion processes and propose self-similar traffic.

Suliman and Lehtomäki assessed the waiting time distribution of the SUs in an interweave CRN in [48]. At the beginning of each time frame, SUs perform spectrum detection on all channels to determine their accessibility. With a service rate corresponding to the size of the time frame, the system is described as a $M/D/1$ queue. The Pollaczek-Khintchine formula is used in the theoretical analysis, which results in the estimation of both the PU and SU queues' waiting times. To evaluate the optimal and incomplete sensing cases, Monte Carlo tests are conducted.

Both PU and SU waiting time in the queue are evaluated in [8], assuming that, high-priority users will have to wait a certain amount of time before trying to access the channel.

The purpose of the delay is to stop SU insufficiency brought on by PU activity. In the preemptive situation, a PU that discovers the channel is already being used by an SU is able to preempt it after its delay, but not in the non-preemptive case. Modelled as a single service unit, a $M/D/1$ queue for both PUs and SUs.

In [56], Zhang et al. evaluate the efficiency of a CRN network with a two-level SU queue. By removing SU packets without transmission that will stay in the network for too long, performance is intended to be improved. The CRN is able to restart priority preemption while it is in interweave mode. The queue consists of two parts: a delay part and a discard part. According to FCFS-based policy, SUs will enter the delay section of the queue if there is no channel available. The number of SUs that may be in the delay queue is fixed by a threshold; any additional ones will be added to the discard portion. SUs produce packets when they are in a queue. The produced packets in the delay queue are buffered at the SU, from which the SUs will be reconnected on an FCFS basis when a band becomes vacant. Those who are waiting to be discarded won't be saved and will be lost. The ratio of packets created over packets lost from an SU is then theoretically evaluated by the authors, using a two-dimensional continuous-time Markov chain (CTMC). With the use of statistical analysis, performance is evaluated by adjusting the queue threshold and the idle periods of the PUs and SUs in the network.

The goal of Oklander and Sidi in [45] is to represent the system dynamics within an interweaving CRN. The stationary probability of a CTMC is determined by the geometric analysis of the matrix. A convenient approach for extracting the stationary probabilities from a transition matrix is provided by methods of matrix geometric analysis. The element of decision-making also includes the estimation of the channel condition.

A queue $M/M/1$ is modelled by Jang and Chang, [33], based on several transmission rates. A fading Nakagami-m band with Doppler switches builds up the system. Analytically, the transmission rate is determined by integrating the Doppler shift and the average fade duration. The exponential operating rate is then compared with the transmission rate. For a priority queue in $M/M/1$, the general equations of delay and throughput are obtained.

The server interruption phenomenon is examined by Azarfar et al. in both single-channel and multiple-channel CRNs, [6]. The queue $M/G/1$ is used to simulate the system. Following a transmission breakdown, scenarios of partial transmission (continue transmission) and retransmission are taken into account. To analyze queue behaviour and determine the performance parameters, the analytical probability is applied. The same authors, in [7], investigate the influence of different queue priority disciplines on the system performance.

The fundamental characteristics of a finite-source queueing model with two (non-independent) frequency channels are addressed in [2] by Almási B. and Sztrik J. Primary and secondary are the two groups into which the users are divided. A newly arriving SU request may utilize the band of the PUs cognitively if the channel of the SUs is already occupied; the licensed channel must be released by the SU, if a PU request arises. Primary requests have priority over those coming from the SUs in the primary channel. A retrial queue (orbit) is connected to the band of the SUs, to which SUs are forwarded if both primary and secondary channels are busy. A MOSEL tool was used to assess the key performance measures.

3.3.2 System model

In this section, we will be having a look at the retrial queuing system that models our cognitive radio network used in this thesis. We are dealing with finite numbers of sources, N_1 and N_2 for primary users and secondary users, respectively. Two non-independent sub-systems make up the network, primary and secondary ones, in which PUs and SUs are operating, respectively. N_1 generate primary requests based on a Poisson-process, with λ_1 an arrival rate and μ_1 a service rate.

The system includes two service facilities: Secondary Channel Service (SCS), which is solely for SUs, and Primary Channel Service (PCS), which is shared by PUs and SUs with a higher priority for PUs over SUs. Both users might start their services right away if the dedicated channel is free.

SUs arrival and service rates are λ_2 and μ_2, respectively. The PUs join a preemptive priority queue if the PCS is already occupied with a high-priority request (PU). Otherwise, the service at the PCS level is paused and the interrupted low-priority request is sent back to the SCS if the PCS is preoccupied with a low-priority request SU.

Generated requests by N_2 in the second sub-network sense the SCS, service might start right away if the unit is free, otherwise, they will be sensing the PCS; if it is empty, service might start opportunistically at the primary service unit, if it is occupied, also, the only option left for the new SU, is to join the orbit, from which, retrial is possible after a random period with rate ν. It should be noted that all the system's inter-event times are supposed to be exponentially distributed. The system's operation is shown in Figure 3.3.

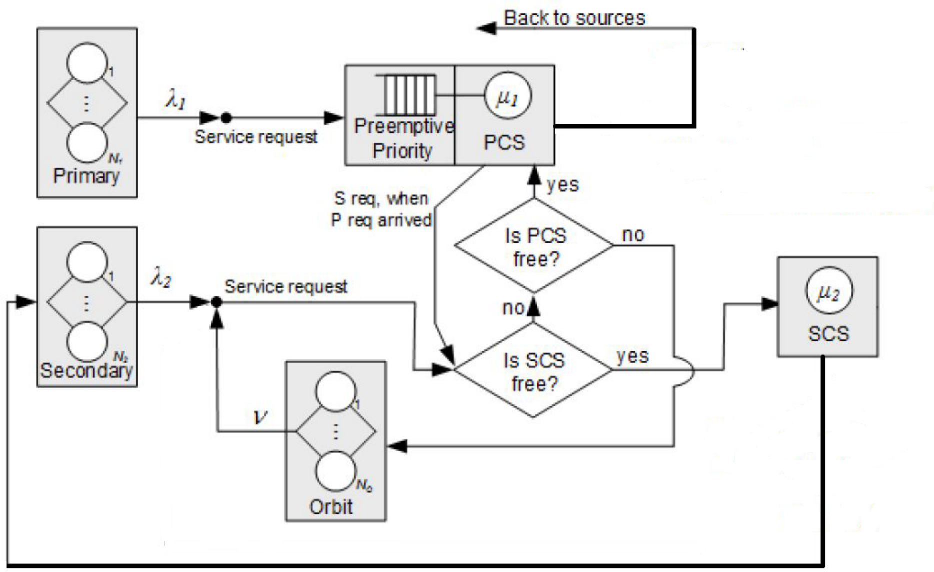


Figure 3.3: A retrial queue model for the cognitive radio network

From these system models, many case studies have been created; in what follows, we list the various situations examined in this thesis.

- We assume that transmission inside the channels is unreliable to accommodate imperfect sensing. Errors may occur during a request's transmission, and the service will fail with a probability of p_1 or p_2 for the PUs and SUs, respectively.

- After a randomly determined retrial period and upon detecting an incoming Primary User (PU), Secondary Users (SUs) persist in sensing both channels without exiting the system until they are allocated by one of the service facilities. Following a First-In-First-Out (FIFO) discipline from the queue, the Primary Users retry.

- The aborted low-priority call at the primary service unit will sense the SCS, and if it is free, the SUs will restore the service from the beginning.
- Balking and reneging concepts are taken into consideration. New coming customers get discouraged upon arrival if the system is crowded (balking); furthermore, they might get impatient and leave the system after joining, after a maximum waiting period (reneging).
- New users are more likely to join the system as it expands in size. This phenomenon referred to as reverse balking, suggests that customers become more encouraged to either wait for or actively enter the system as it grows. The nature of this growth, whether in features, capacity, or popularity, plays a crucial role in influencing user behavior.

3.4 Simulation of finite-source cognitive radio network

This section demonstrates how to develop a simulation program that simulates the finite-source, retrial queueing cognitive radio network system shown in Figure 3.3.

3.4.1 Problem statement

Consider primary and secondary servers that are consisting a retrial queueing system, in which arrival times $A_{11}, A_{12}, ..., A_{1n_1}$ and $A_{21}, A_{22}, ..., A_{2n_2}$ for the primary and secondary users are independent random variables with the same distribution. A user who comes and discovers the relevant server in an idle state, instantaneously starts the service, the service times of the subsequent main and secondary customers are independent random variables with identical distributions as $S_{11}, S_{12}, ..., S_{1n_1}$ and $S_{21}, S_{22}, ..., S_{2n_2}$. The arrival of a new PU with a busy server result in joining the FIFO queue. The unit makes a FIFO selection of a client from the queue after finishing the main service. The retry times $R_1, R_2, ..$ are likewise independent and identically distributed random variables, and the secondary customers who discover both channels occupied enter the orbit.

When no customers are in the system and the server is idle, the simulation starts at time 0 with generating random arrival times for each PU and SU $A_{11}, A_{12}, ..., A_{1n_1}$, $A_{21}, A_{22}, ..., A_{2n_2}$ with n_1 =: *Primary number of sources* and n_2 = *Secondary number of sources* while scheduling the "*Arrival event*" after each generation. The initial arrival event will take place after the shortest inter-arrival time, which is $\min(A_{11}, A_{12}, ..., A_{1n_1}, A_{21}, A_{22}, ..., A_{2n_2})$. Up until N number of customers have finished their service, we would like to simulate this system. When the nth client joins the service, the simulation will end.

3.4.2 Logic and organisation of the program

The components for the program to simulate our retrial queueing CRN are presented in this subsection. SimPack, a set of C and C++ libraries and executable programs for computer simulation, provides the foundation for the developed simulation. This collection of simulation techniques includes discrete event simulation, continuous simulation, and combined (multimodal) simulation. This was addressed by Paul A. Fishwick in [19], in 1992. Involving various input parameters, to analyse the characteristics of the system, we build a simulation model. The determination of performance measurements and the provision of accurate equations, however, might be challenging and complex in particular circumstances.

The operation of the systems can be characterized by the utilization of a continuous-time Markov chain in specific scenarios where all inter-event intervals follow exponential distributions, allowing for the computation of primary stationary performance metrics. Otherwise, due to the extensive state space of the associated Markov chain, calculating system measurements or solving steady-state equations becomes challenging. Consequently, the most effective approach for obtaining approximate results in such complex scenarios is through the simulation of the system model.

To make this process easier, some software packages have been created, such as those by [22, 23, 24, 31] which are capable of modelling and assessing complicated systems, based on exponential distributions. Defining the fundamental events and updating the system state each time an event takes place, is the most crucial phase

The time interval between two successive events does not modify the system's status. Each basic event is connected to a timer that will keep track of the moment at which the event happens, to include the essential events in the simulation model. After an event has been processed, the simulation model rejoins all potential future events and selects the one with the lowest clock value.

Figure 3.2 and subsection 3.2.1 define the simulation components.

We implemented in our simulation environment multiple distributions for random values generation, including exponential, hyper-exponential, hypo-exponential, lognormal, gamma and Pareto. Both Herb Schwetman in [47] and M. H. MacDougall in [39] served as inspiration for this collection of routines. The simulation program also includes routines for startup, timing, and report production in addition to the main program and random number generators. However, the most significant activity occurs during the events' preparations, which we rank as follows:

- Customer arrival to the system:
  As it sounds, the occurrence of the event follows straight away the arrival of PU or SU.

- Customer arrives at the PCS:
  This event is the result of an idle PCS upon arrival of PU or SU, and the service is not interrupted by the entrance of a cognitive user or a server breakdown. SUs may arrive from the orbit repeatedly or the primary users directly from the source thanks to cognitive radio channels.

- Customer arrives at the SCS:
  If a new arriving SU finds SCS free and no service is interrupted by another arrival or server's breakdown, this event will be scheduled.

- Customer departure after job completion at PCS:
  If a SU or PU is done with service in the primary service unit, this event will be launched.

- Customer departure after job completion at SCS:
  If a SU is done with service in the secondary service unit, this event will be launched.

- Arrival from the orbit:
  When a secondary user (SU) finds both the primary (PCS) and secondary (SCS) service units busy upon arrival, or if the SCS unit goes down during service, this event is triggered.

Typically, in simulation, clocks are represented by real numbers. Therefore, it is not possible to have multiple events happening at once. We have integrated the aforementioned network models and scheduling algorithms into the simulation program, and the basic procedures for user arrival and departure are shown in the following figures.

A flowchart for the arrival procedure may be seen in Figure 3.4. To start, using PCS and SCS, all the upcoming arrivals' future times are calculated and stored in an events list.

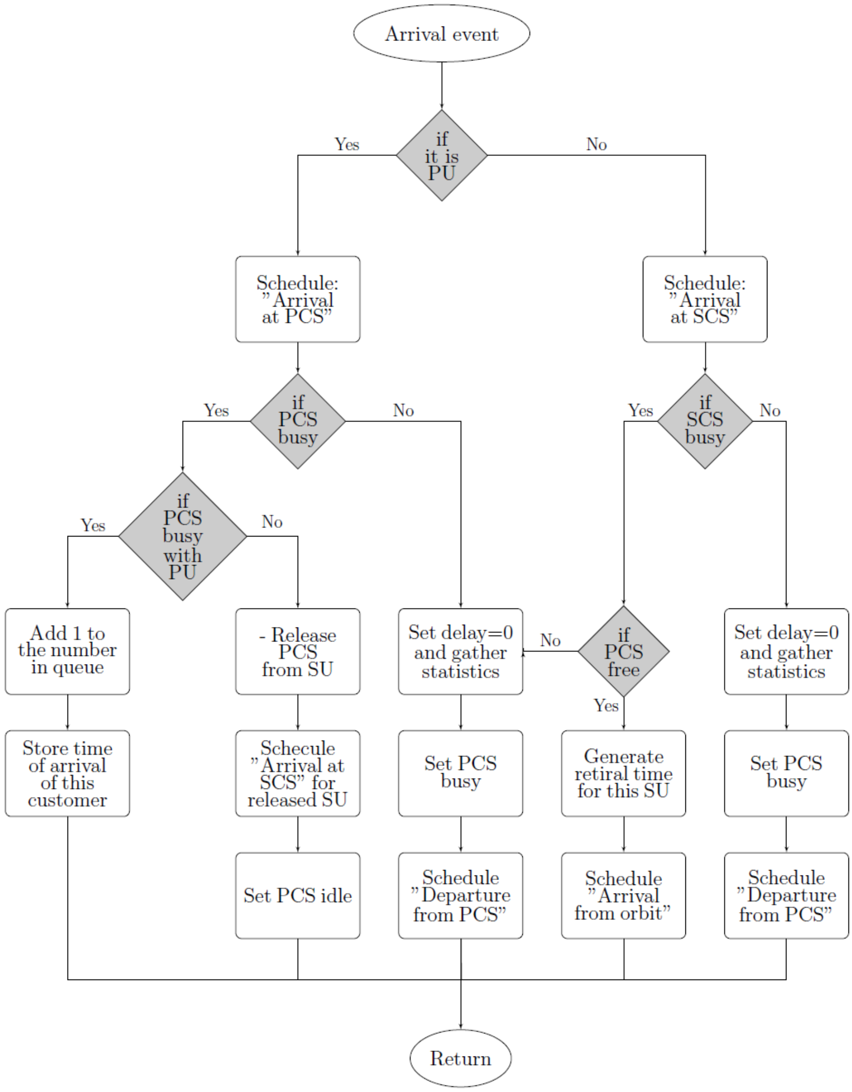


Figure 3.4: Flowchart for arrival routine

Afterward, two checks are performed to ensure that a Primary User (PU) is detected and that the Primary Channel Status (PCS) is free. If both conditions are met, the service begins. The PCS status changes to 'busy,' and the departure time of the newly arrived PU is added to the event list. In the 'busy' status, if a newly arriving PU detects a Secondary User (SU) in the PCS, the PU must be released, and a new arrival event is scheduled at the Secondary Channel Status (SCS) to allow the PU to commence.

Table 3.1: Event description in the simulation program

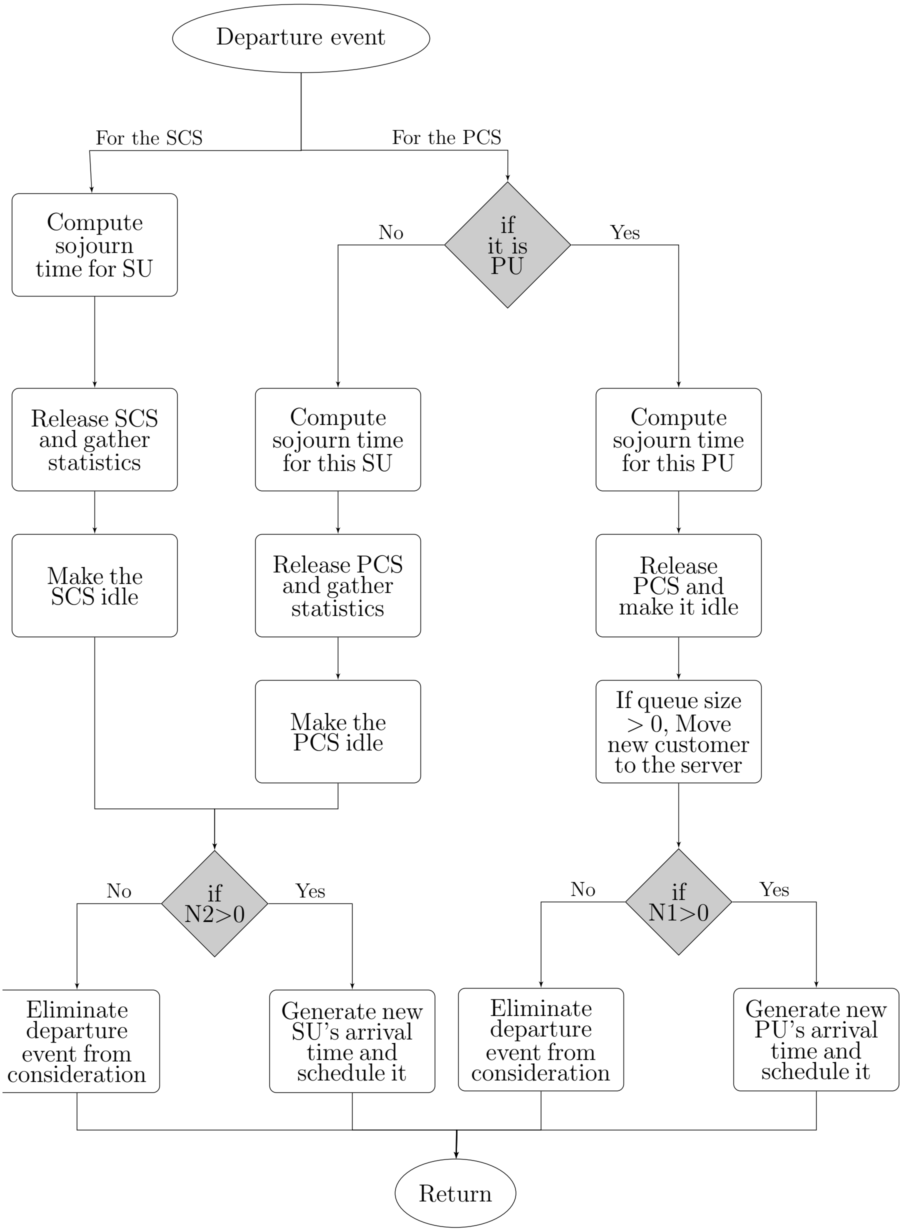
Events	Event defined as
Customer arrival	1
Arrival from orbit	2
Request the PCS	3
Request the SCS	4
Departure of customer:	
- Departure from the PCS	5
- Departure from the SCS	6

If the PCS is busy processing another PU, the arrival time of the newly licensed user is recorded, and the user is placed at the end of the First-In-First-Out (FIFO) queue.

For arriving SUs, they check whether the secondary channel is busy. If occupied, the SU rechecks the PCS. If none of the service units is occupied, the service is initiated by the arriving secondary customer. After a random service time, a departure from the Secondary Channel Status (SCS) is planned. A new retrial time is generated for the arriving SU, and if none of the servers is available, the arrival from orbit event is scheduled and added to the event list.

Figure 3.5 illustrates the departure event which is triggered from both PCS and SCS when a service completion occurs. The server goes to an idle state if a PU leaves no other users in the FIFO queue behind. If the queue is still containing users, preemptive access is taken place. As a result, the queue length is reduced by 1, and the wait time for that customer is computed and logged in the relevant statistics counter.

When a PU arrives at the PCS level and interrupts a secondary task, the interrupted job's departure from the PCS event is no longer taken into account. However, exit from SCS happens after a SU completes their service. The secondary server is not in use, and the customer's time in orbit is calculated and recorded as waiting time in the relevant statistics counter. If and only if the primary or secondary number of sources (N_1 or N_2) is larger than 0, a new primary or secondary arrival time is produced and added to the event list in both scenarios.



N_1: Primary number of sources
N_2: Secondary number of sources

Figure 3.5: Flowchart for departure routine

3.4.3 Confidence interval, estimation of the mean and variance

The following will help us to discover some fundamental formulas that have been useful to calculate and estimate the mean and variance during simulation modelling, [37] is a good example of this.

Assuming that $X_1, X_2, ..., X_n$ are random independent observations with finite-source mean μ and variance σ^2; the sample mean is given by:

$$\bar{X}(n) = \frac{\sum_{i=1}^{n} X_i}{n} \tag{3.21}$$

it is an impartial estimator of μ. If we do a vast number of separate tests, each resulting in an $\bar{X}(n)$. The average of $\bar{X}(n)$ will be μ. Similarly, the sample variance is given by:

$$S^2(n) = \frac{\sum_{i=1}^{n} [X_i - \bar{X}(n)]^2}{n-1} \tag{3.22}$$

it is an impartial estimator of σ^2, since $E[S^2(n)] = \sigma^2$.

Without any further information, using $\bar{X}(n)$ as an estimator of μ presents a challenge since there is no way to determine how close $\bar{X}(n)$ is to μ. Given that $\bar{X}(n)$ is a random variable with a variance of $Var[\bar{X}(n)]$, $\bar{X}(n)$ may be close to μ in one experiment while deviating significantly from m in another. The standard method for evaluating the precision of $\bar{X}(n)$ as an estimator of μ is to build a confidence interval.

We start with the traditional central limit theorem in order to build the confidence range for μ,

By assuming the following random variables $X_1, X_2, ..., X_n$ having a finite mean μ and a finite variance σ^2, and in order for us to construct σ^2 (the confidence range), We begin with the typical central limit theorem.

Let Z_n be the random variable $[\bar{X}(n) - \mu]/\sqrt{\sigma^2/n}$ and let $F_n(z) = P(Z_n \leq z)$ be the distribution function of Z_n for a sample size of n.

In essence, the theorem states that regardless of the distribution of X_i's, the random variable Z_n will be roughly distributed as a standard normal random variable if n is big enough. The sample mean $\bar{X}(n)$ is roughly distributed as a normal random variable with mean μ and variance σ_2/n, as can also be shown for high n. However, this still holds true if we substitute σ_2 for $S^2(n)$ in the formula for Z_n since the sample variance $S^2(n)$ converges to σ^2 as n grows larger. As a result of this modification, the theorem now states that if n is large enough, the random variable $t_n = [\bar{X}(n) - \mu]/\sqrt{S^2(n)/n}$ is approximately distributed as a standard normal random variable. For big n, it follows that:

$$
\begin{aligned}
&P(-z_{1-\alpha/2} \leq \frac{\bar{X}(n) - \mu}{\sqrt{S^2(n)/n}} \leq z_{1-\alpha/2}) \\
&= P[\bar{X}(n) - z_{1-\alpha/2}\sqrt{\frac{S^2(n)}{n}} \leq \mu \leq \bar{X}(n) + z_{1-\alpha/2}\sqrt{\frac{S^2(n)}{n}}] \\
&\approx 1 - \alpha
\end{aligned}
\tag{3.23}
$$

Where $z_{1-\alpha/2}$ (for $0 < \alpha < 1$) is the upper $1 - \alpha/2$ critical point for a standard normal random variable. Therefore, if n is sufficiently large, an approximate $100(1 - \alpha)$ percent confidence interval for μ is given by:

$$
\bar{X}(n) \pm z_{1-\alpha/2}\sqrt{\frac{S^2(n)}{n}} \tag{3.24}
$$

where for a given set of data $X_1, X_2, ..., X_n$ the lower confidence interval endpoint is $\bar{X}(n) - z_{1-\alpha/2}\sqrt{S^2(n)/n}$ and the upper confidence interval endpoint is $\bar{X}(n) + z_{1-\alpha/2}\sqrt{S^2(n)/n}$.

3.4.4 The batch mean method

The statistics module class is based on an adaption of Andrea Francini's statistics package in [20], which was introduced in 1994. A statistical analysis technique that can perform a quantitative approximation of the mean and variance values of the observed variables is the statistic class. The challenge with employing a mean estimator in the absence of any additional data is that there is no way to determine how near the estimator is to the mean. On one analysis, the approximation could be close to the mean whereas on another, it might deviate significantly. Building the confidence interval for this latter is a frequent technique used in simulation to assess the accuracy of the estimator for the mean. Equation 3.21 estimates the mean value of a generic variable X, and subsection 3.4.3 expresses the idea of the confidence interval and the confidence level mathematically.

The batch means method, which divides the observations gathered into sequential blocks of data, or "batches", is used to estimate the mean and variance values and build up its confidence interval followed by considering the means derived from these batches as independent. It is the most often used confidence interval approach for a steady-state simulation's output analysis, [53, 10, 12, 18] provide more information on this methodology.

A stochastic process can only be statistically estimated after it has already attained the statistic stationary condition. Since the observations made during the early transitional period may significantly depart from the accurate estimate values, they must be ignored. The "warm-up phase" refers to this time frame, See [51].

Let M be the duration of the simulation, for instance, and K be the number of observations at the start of the simulation (warm-up period). The relevant run, of length $M - k$, is split into N batches, with $n = \frac{M-K}{N}$ observations in each batch.

Two approaches to figuring out the initial transient are suggested by Andrea Francini:

- a procedure that includes allocating the three parameters $n0$, N and σ. If the k averages have an accuracy of σ, the warm-up phase may be thought of as having ended after N_0 batches of data, given a sequence of averages $\bar{X}1(n_0), \bar{X}2(n_0), ...,$ pertaining to batches of successive data.

- In the statistical analysis of the simulation results, the averages associated with a set of successive batches of data, $\bar{X}_1(m), \bar{X}_2(m), ...$ are utilized as secondary output data. The following is the final estimate for the expectation $\bar{\bar{X}}$:

  $$\bar{\bar{X}}(N, n) = \frac{1}{N} \sum_{i=1}^{N} \bar{X}_i(n) = \frac{1}{nN} \sum_{i=1}^{N} \sum_{j=1}^{n} X_{ij}(n) \tag{3.25}$$

  where n is the size of the batches and N is the number of batches (number of observations in one batch).

  An estimate of the variance of a single $\bar{X}_i$ is then given based on their sample variance:

  $$S^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\bar{X}_i(n) - \bar{\bar{X}}(n))^2 \tag{3.26}$$

Knowing what "n sufficiently big" refers to is challenging when applying equation 3.24 in subsection 3.4.3 to get the lower and upper confidence-interval endpoints for $\bar{\bar{X}}$. The coverage of a targeted $100(1-\alpha)$ percent confidence interval will often be less than $1-\alpha$ if n is selected too small. This is the reason why 3.24 states that the confidence interval is simply an approximation. Instead, the confidence interval for the mean $\bar{\bar{X}}$ (at confidence level $1-\alpha$) is instead expressed as follows:

$$\bar{\bar{X}}(N,n) \pm t_{N-1,1-\alpha/2}\frac{S}{\sqrt{N}} \tag{3.27}$$

where t_N has a student t distribution with $N-1$ degrees of freedom (df).

After considering the aforementioned comments, we ran a number of simulations to examine the performance measures of the retrial queue cognitive radio network with a limited number of sources. The following settings were used:

- The confidence interval's relative half-width must be at least 0.05 in order to terminate the simulation (when all of the evaluated processes reach the chosen accuracy level, the run is terminated).
- A minimum of 5000 observations must be gathered before the first transitory shutting condition is verified.
- 10000000 is the maximum number of treated observations.
- 30 batches were utilized to test the first transient duration.
- The first transient closure condition is checked using 10 transient batch means.
- It takes 0.99 accuracy to shut the first transient detection.
- The batch size employed initially for the stationary analysis is 10 000. (The means are linked, and the batch size doubles, if the number of gathered batch means exceeds the designated memory space that is available).
- 95% is the confidence level.

ht

Table 3.2: Event description for unreliable servers in the simulation program

Events	Event defined as
Failure of primary server	7
Failure of secondary server	8
PCS operates	9
SCS operates	10

3.5 Simulation of non-reliable servers in finite-source cognitive radio network

Several case scenarios could be applied to our system; in this part, we present one of them. We assumed that the SCS is unreliable and prone to failures and repairs according to random distributed times. In order to simulate the performance of our system model, additional routines for the events are added to the simulation.

During the simulation run, the following new routines may happen in addition to the circumstances indicated in subsection 3.4.2 and table 3.1:

1. SCS failure: The operating time the unit needs to be in service and functioning is generated.
2. SCS repair: When this event is triggered, a random time in the future is created to determine when the secondary server will fail. During this period, the server is operational.

The simulation model of a finite-source retrial queueing model with unreliable servers is shown in Figure 3.6, After the initialization function that assigns the program's parameter settings, the simulation begins by creating N_1 and N_2 random times for the customers who are coming and storing them in the list of events. Following that, a client arrival event takes place (the customer having the smallest arrival time). The user will then be checked to see whether they are the PU. If this is the case, this user must sense the channel to determine if the PCS is busy or down, and in this instance, the user moves up to the front of the FIFO queue. The service begins at the main unit if the PCS is up and running and not otherwise in use. In contrast, if the user is secondary, a check is made to see if the SCS is operational or occupied, if not, SU can sense the PCS as well, as allowed by the cognitive technology. Similar to the previous approach, retrial events happen from the orbit if the PCS is busy. It

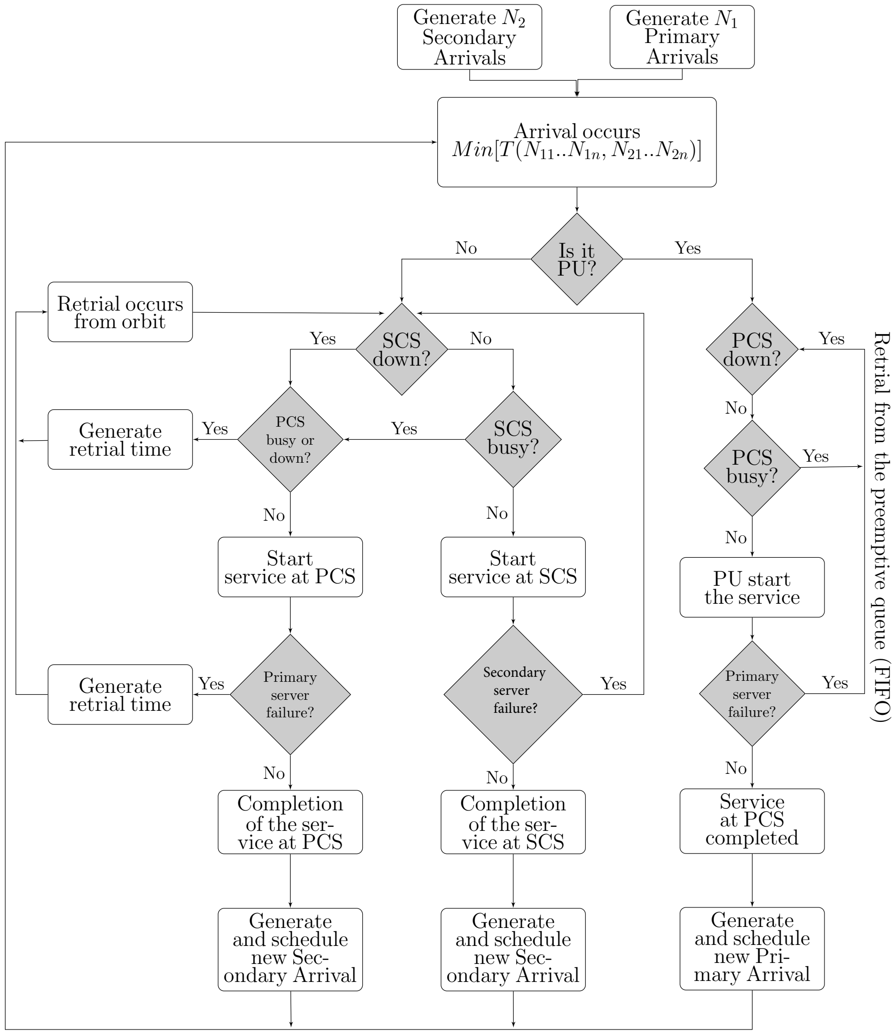


Figure 3.6: Simulation model of non-reliable servers

should be noted that secondary service units may fail during a service period and customers will then retry the service from the orbit or queue.

[43] addressed non-reliability on CRN, assuming that both servers are subject to random break-downs repairs.

3.6 Simulation methodology

Foremost, this PhD report utilized various software tools for the simulation. The integrated development environment employed was NetBeans IDE version 8.0. The simulation program, implemented in the C programming language, utilized the C compiler GCC and the debugger GDB within the Cygwin environment. A crucial element of the simulation tool was SIMPACK, a package created by Paul A. Fishwick in 1990. SIMPACK comprises a set of routines specifically designed for discrete event simulation, aiding in tasks such as initializing data structures, scheduling events at specified times, and more. Significantly contributing to the simulation was the 'stat-mod.c' class, developed in 1998 by Andrea Francini, Marco Mellia, Giorgio Politano Gabriele Favalessa, and Luca Medico. This class encompassed functions responsible for managing the statistical aspects of a simulation run, including procedures for estimating mean values and variances using the batch mean method. Moreover, in specific scenarios addressed in this thesis, the generation of event times relied on the Gamma distribution, achieved through the use of GLS libraries.

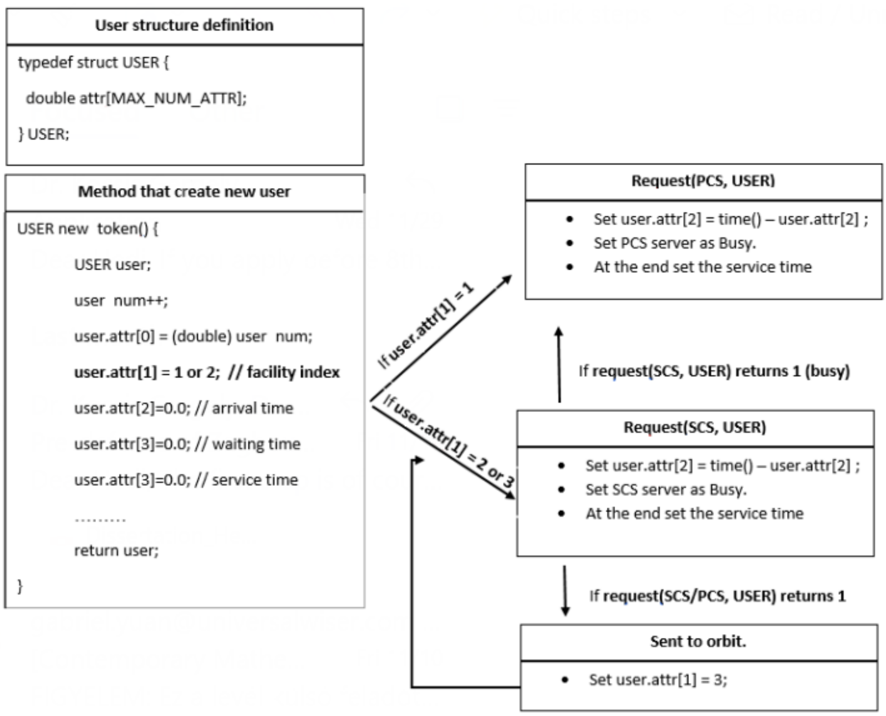


Figure 3.7: Users requirements and uses cases

The simulation software users were defined as data structure that contains specific attributes, including:

- Unique user ID
- Request facility index (Primary or Secondary)
- Additional attributes such as arrival time, waiting time, and service time were represented as variables in which we recorded the respective event times.

These data structures were randomly generated for primary (facility index = 1) and secondary (facility index = 2) users, following a generally distributed inter-arrival time.

The request-facility function was developed to handle different use cases. It takes the server-id (PCS or SCS), the user data structure, and priority as input parameters.

The function verifies whether the PCS or SCS should be used based on the facility index value stored in the user data structure. If the facility index is 1 (primary), the request-facility function utilizes PCS as an input parameter. On the other hand, if the facility index is 2 (secondary), the request-facility function employs SCS as an input parameter.

Depending on the state of the selected server, if it is busy, another request-facility function will be called, this time using PCS as an input parameter. The flowchart in Figure 3.7 explains more the user requirements and uses cases in the simulation.

3.7 Conclusion

This chapter was broken down into three key subsections to help readers understand how our cognitive radio network was simulatively modelled using queuing theory. Two service facilities co-exist, a first primary sub-network that is linked to a preemptive queue and has a licensed frequency band that is not congested. The second component is a retrial queueing sub-network with an overloaded frequency band. The second part's queue is symbolized by an orbit from which packets retry their requests based on a distributed random time. While their service unit is occupied, the secondary packets may cognitively since the main channel since the two subsystems are not independent of one another. This chapter begins by providing an overview of the random processes and distributions that are relevant to our investigation. The studies that were utilized to model the cognitive radio networks from the literature are then presented in the second section of the chapter. As a result, we've highlighted our simulation modelling's key features and parameters in the chapter's last section. The results of the simulation program are presented in the next chapter.

4

Performance evaluation and analysis

With the use of figures, this chapter demonstrates the various simulation outcomes attained throughout this study project.

Contents

4.1 Introduction 50
4.2 Unreliability Analysis of finite-source cognitive radio 50
4.2.1 System model 51
4.2.2 Validation of results 52
4.2.3 Numerical results 53
4.3 Finite-source cognitive radio network with unreliability and abandonment 67
4.3.1 System model 68
4.3.2 Validation of results 70
4.3.3 Numerical results 70
4.4 Analysis of finite-source cognitive radio network with balking and reneging 79
4.4.1 System model 80
4.4.2 Validation of the results 81
4.4.3 Numerical results 81

4.1 Introduction

As shown in Figure 4.1, In this chapter, we introduce a finite-source queuing model with two non-independent frequency channels. According to the CRN modelling, users fall into two categories: Primary Users (PUs) have a licensed frequency that is not overloaded. Secondary Users (SUs) who have access to a frequency band may experience overloading.

If the secondary frequency band SUs is engaged, a newly incoming SU request may utilize opportunistically the primary band. However, releasing the non-licensed frequency when a PU request comes is a must for SUs. The primary band in our system is modelled by a FIFO queue, with PU requests having pre-emptive priority over SU requests.

We aim to study the impact of the inter-event times distributions and their parameters on several performance measures, mainly the mean and variance of the PUs and SUs' response times, based on several study scenarios of the developed model.

We have implemented multiple distributions, trying to investigate their effect on the response times, using simulation. The obtained results were illustrated in multiple figures.

The verified new investigation results presented in this chapter are included in the journal and conference publications listed below [J1[i], J2, J4, J5, C3, C4, C6, [ii]].

4.2 Unreliability Analysis of finite-source cognitive radio

In this section, we introduce the first case scenario that we applied to the system. We evaluate the performance of a cognitive radio network using a queuing model illustrated in Figure 4.1 with a secondary server subject to random breakdowns and repairs.

The service of PUs and SUs combines two linked sub-systems, primary and secondary, respectively. The main sub-system is given a FIFO queue, while the secondary sub-system (SCS) is given an orbit. Both units have a finite number of sources, N_1 and N_2 in charge of generating the requests.

[i] "J" denotes Journal papers
[ii] "C" denotes Conference papers

The primary service unit (PCS) is the destination of the newly created high-priority packets. If the unit is not in use, the packet's service starts right away. The packet enters the pre-emptive priority queue if the server is preoccupied with a high-priority request. The service is stopped if the later unit is occupied by a low-priority request, consequently, the PU starts service and the aborted SU, will be routed back to the SCS. The interrupted job is sent to either the server or the orbit depending on the condition of the secondary channel.

Similarly, low-priority requests are headed to the SCS, if they sense an idle unit, the service begins; otherwise, the packet will be headed to the PCS trying to opportunistically use this unit if it is free, if not, the only option left for the new arriving SU is to join the orbit. After a random retrial time, SUs will be trying to get service again, from the orbit.

The secondary unit of our system is assumed to be unreliable; it is subject to random breakdowns and repairs during both busy and idle states. The failure (operating) and repair times are, presumptively, non-exponentially and generally distributed, using Gamma, Pareto, Log-normal, Hypo-Exponential, and Hyper-Exponential.

It is assumed that all random times used in the model's construction are independent of each other. We intend to use simulation to examine the impact of the second server's unreliability distributions on the key performance measures of the system.

> **New results in (J1, J5, C4, C6)** *We dealt with a finite-source cognitive radio network in which the low-priority subsystem is unreliable and subject to random breakdowns and repairs that occur in busy and idle states. Our goal is to use simulation to analyse the impact of the servers' unreliability time distributions and failure appearance (idle or busy status) on the key performance measures.*

4.2.1 System model

As mentioned above, N_1 will be generating high-priority tasks according to exponentially distributed inter-arrival times with the parameter λ_1. The service times of PUs are assumed to be also exponentially distributed, with the parameter μ_1.

The second sub-system, inter-arrival and service times are generated exponentially with a mean value N_2/λ_2 and parameter μ_2, respectively.

From the orbit, low-priority jobs will be trying to get service again after an exponentially distributed time with a parameter ν/N_2.

Operating and repair times responsible for the breakdowns and repairs on the secondary server are random variables, generally distributed using: Exponential, Hypo-Exponential, Gamma, Log-normal, and Pareto distributions. The following rates refer to the unreliability part:

Failure rate while idle: θ_2,
Failure rate while busy: γ_2,
Repair rate: σ_2.

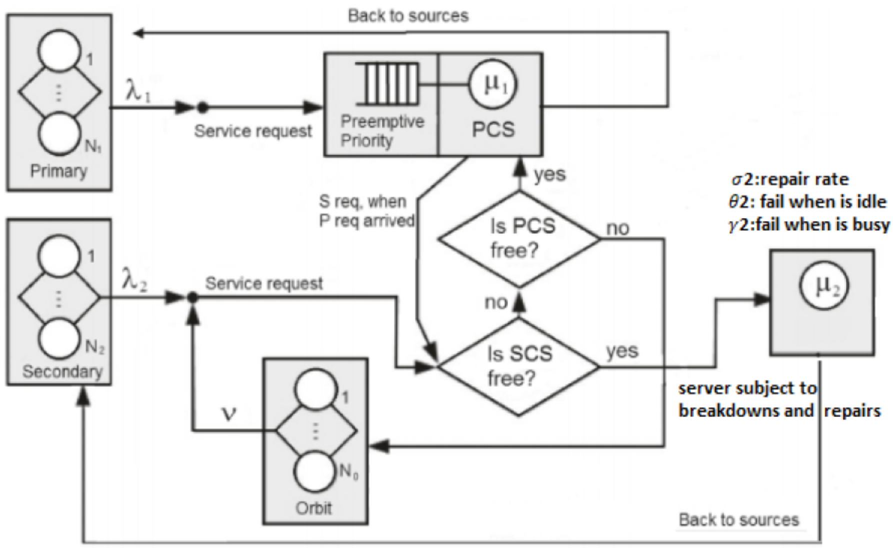


Figure 4.1: Finite-source retrial queueing system: Modelling the Cognitive Radio Network with non-reliable SCS

4.2.2 Validation of results

The results of this case were obtained using a simulation program that implemented the "batch-mean" method.

To generate point and interval estimators, this approach seeks to gather a number of independent samples (batches) by accumulating several contiguous simulated observations. Each batch size must be sufficiently large to prevent excessively correlated sample averages (A commonly used rule of thumb is to use a minimum of 30 batches, with each batch containing at least 10 to 20 samples and having 0.05 as accuracy level, however, the optimal number of

Table 4.1: Confidence intervals of the figures

Fig.	Obs. Point	Distubition	N	$t_{N,1-\frac{\beta}{2}}$	95% Confidence Int.	
					LB	UB
4.2	1.5	Pareto	68	1.995	6.087	6.633
		Gamma	70	1.994	6.847	7.113
		Hyper	65	1.997	6.212	6.895
		Lognormal	60	2.000	6.230	6.903
4.8	1	Pareto	86	1.988	6.201	6.85
		Gamma	85	1.988	7.745	8.287
		Hyper	79	1.990	6.242	6.836
		Lognormal	90	1.987	6.251	6.829
4.12	2	Pareto	105	1.960	8.191	8.782
		Gamma	95	1.985	7.591	8.187
		Hypo	83	1.989	8.189	8.789
		Lognormal	89	1.987	8.29	8.867

batches and sample sizes can vary depending on the specific application). The final mean or variance is then calculated by averaging the data points from each batch. More details on this approach could be found in [12].

The associated theorem may be used to generate the confidence interval 4.1 for the method given above, as shown in [14].

The distribution's confidence intervals are displayed in Table 4.1.

$$\hat{\mu}N \pm t_{N,1-\frac{\beta}{2}} \frac{S}{\sqrt{N}}, \tag{4.1}$$

with confidence interval $1-\beta$.

- $\hat{\mu}N$: Estimator for the mean response time,
- N: Number of batches,
- $t_{N,1-\frac{\beta}{2}}$: The $1-\frac{\beta}{2}$ critical value of the Student t distribution with N degrees of freedom
- S: Sample standard deviation.

4.2.3 Numerical results

The impact of the failure and repair time distributions having the same mean but different variances are shown in this subsection through several figures. See

Table 4.2: Numerical values of model parameters

N_1	N_2	λ_1	λ_2 / N_2	μ_1	$\mu 2$	ν / N_2	θ_2	γ_2	σ_2
10	10	0.01	x-axis	1	1	0.01	0.1	0.1	1

4.2 for the input parameter's numerical values. We outline a few aspects of our system that should be kept in mind:

- When the steady-state equations are not solvable, in the case of non-exponential distributions (unlike exponential), the simulation approach is the most efficient for performance modelling and analysis.
- We assume that within our system, the secondary service that was interrupted by the arrival of PUs in the PCS or by a server failure in the SCS, will be restored from scratch (non-intelligent). Additionally, the system won't be obstructed by the service unit's failure, and free sources will continue to generate new jobs.
- In order to study the effects of the operating and repair times distributions on the behaviour of the system, we ran many simulations. Specifically, we have looked at scenarios when the server goes down in an active or inactive condition, independently.

We have chosen the three case scenarios, among many that our simulation may address:

- **Scenario 1**: The operating time is Exponentially distributed and the SCS repair time is generally distributed,
- **Scenario 2**: SCS operating time is generally distributed when the server fails during idle state, the repair time is Exponentially distributed,
- **Scenario 3**: SCS operating time is generally distributed when the server fails during a busy state, the repair time is Exponentially distributed.

4.2.3.1 Repair time is generally distributed

First, having equal means and variances, our goal was to determine how the repair times distributions affect the system's performance. We decided to split the examination into two parts according to the square coefficient of variation C_x^2 when greater or less than one. The input variables for the distributions of the repair times are shown in Table 4.3.

Table 4.3: Parameters of the distribution

Distribution		Hyper	Hypo	Gamma	Pareto	Lognormal
Fig 5,6,7	Mean	N/A	1	1	1	1
	Variance	N/A	0.68	0.68	0.68	0.68
	Parameters	N/A	λ_1=1.25 λ_2=5	α=1.470 β=1.470	α=2.751 K=0.611	m=0.720 σ=-0.259
Fig 2,3,4	Mean	1	N/A	1	1	1
	Variance	2.56	N / A	2.56	2.56	2.56
	Parameters	λ_1=0.661 λ_2=1.3380 p=0.330	N/A	α=0.3906 β=0.390	α=2.179 K=0.541	m=1.126 σ=-0.634

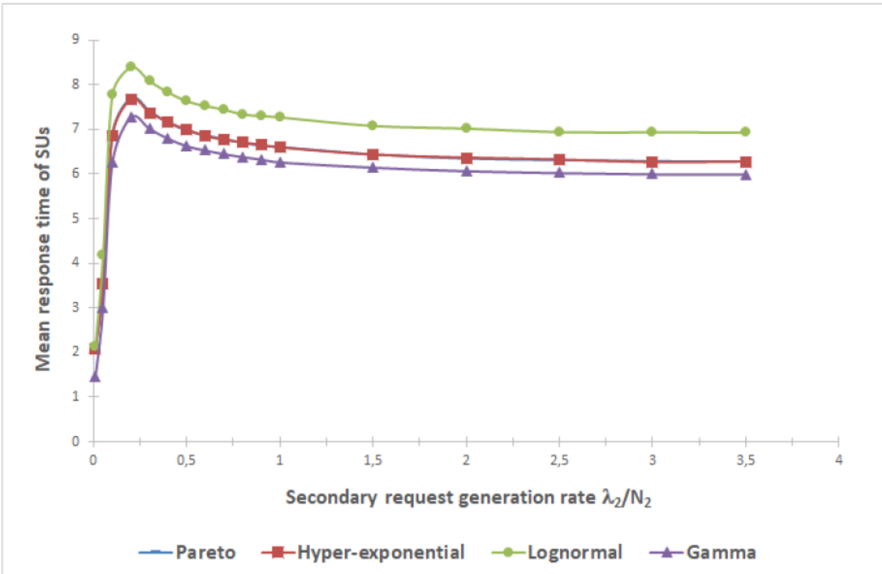


Figure 4.2: The effect of repair time distributions on the mean sojourn time of cognitive users vs secondary arrival rate

Squared coefficient of variation C_x^2 is greater than one The effects of the server's repair time distribution on the average sojourn time, the overall server utilization, and the average service time of secondary customers are shown, respectively, in Figure 4.2, 4.3 and 4.4, these findings were generated in relation to the secondary arrival intensity.

The figures demonstrate how the repair times distribution has a significant impact on the characteristics. Gamma distribution in Figure 4.2, for instance, offers the shortest value of the mean response time, whereas the log-normal distribution provides a greater value of the mean. Moreover, since the gamma

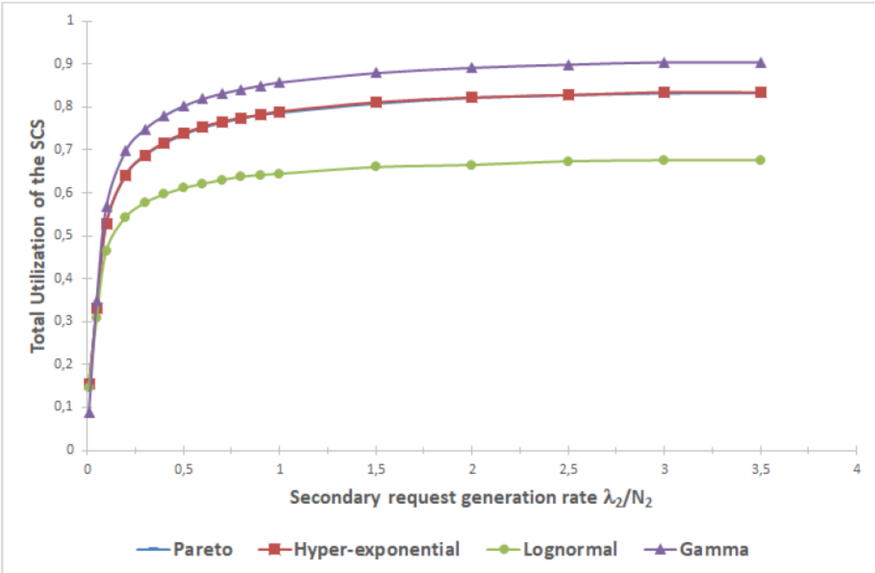


Figure 4.3: The effect of repair time distributions on total utilization of the secondary server vs secondary arrival rate

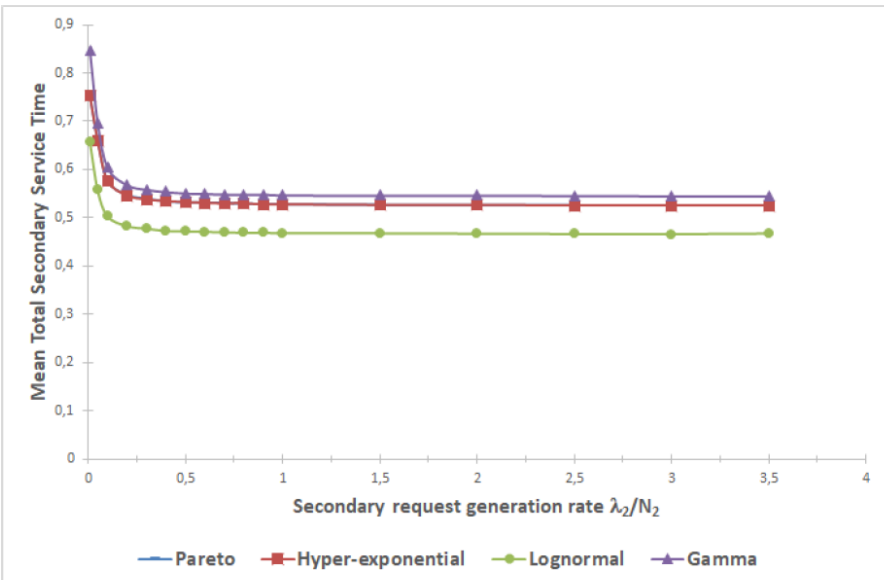


Figure 4.4: The effect of repair time distributions on total utilization of the secondary server vs secondary arrival rate.

distribution produces low values using the aforementioned input parameters, the server recovers more quickly. As a result, Figure 4.3 shows a higher value of utilization. Similar to Figure 4.3, the gamma distribution in Figure 4.4 provides the largest value of the mean service time since the system is frequently in operating mode.

The observed random variable may affect how our results are interpreted. For example, if the distribution's mean and variance are modified, the value of the mean total of primary and secondary service time will vary. We may investigate characteristics that are almost impossible to analyse analytically by simulating the performance measures of such a system. However, the following is how we explain the outcomes:

The repair time is generally distributed in Figure 4.2. The relative probability of the random variable x (repair time) is greater in the case of the log-normal distribution than in the case of the gamma, which means that the repair time will most likely take greater values in the case it is log-normally distributed, as can be seen from the graph of the Probability Density Function of the lognormal and gamma distributions. Consequently, users must respond to requests more quickly since server repairs take longer. On the other hand, Figure 4.3 demonstrates that the server's utilization is lower when the repair time is lognormally distributed than when it is gamma distributed since the server is often down and unoccupied.

Squared coefficient of variation C_x^2 is less than one In this subsection, we investigate and compare the impact of the repair time distribution on the same features chosen above. In this case, however, the hyper-exponential distribution is swapped out for the hypo-exponential distribution, and new parameters are set so that their coefficient of variation is less than one, as seen in Table 4.3. Similar results of the repair time distribution on the average response time/service time of secondary requests and the use of SCS vs secondary inter-arrival rate are seen in Figures 4.5, 4.6 and 4.7. The values of the performance metrics vary across two sets of distributions in this instance, when the distribution's C_x^2 is less than 1. Hypo-exponential and Pareto distributions provide estimates with comparable values. These values exceed the ones that the log-normal and gamma distributions produced. Similar to the case of $C_x^2 > 1$, the explanation of the displayed influence of the distributions follows the same pattern.

In this case, the anticipated phenomenon was observed, including the property of having a maximum mean value. Additionally, when the arrival intensity is increased, the server is used more often and the mean service time decreases.

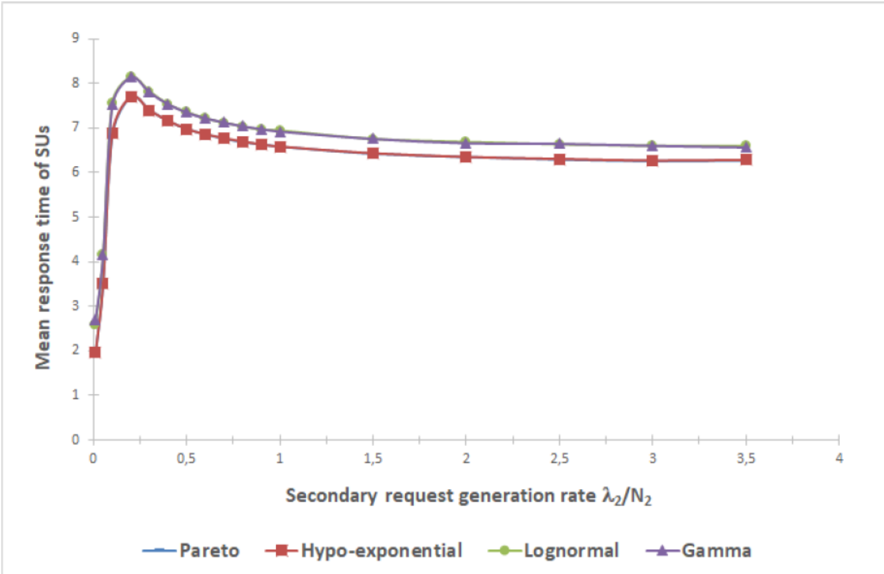


Figure 4.5: The impact of repair time distributions on the mean response time of SUS vs λ_2/N_2

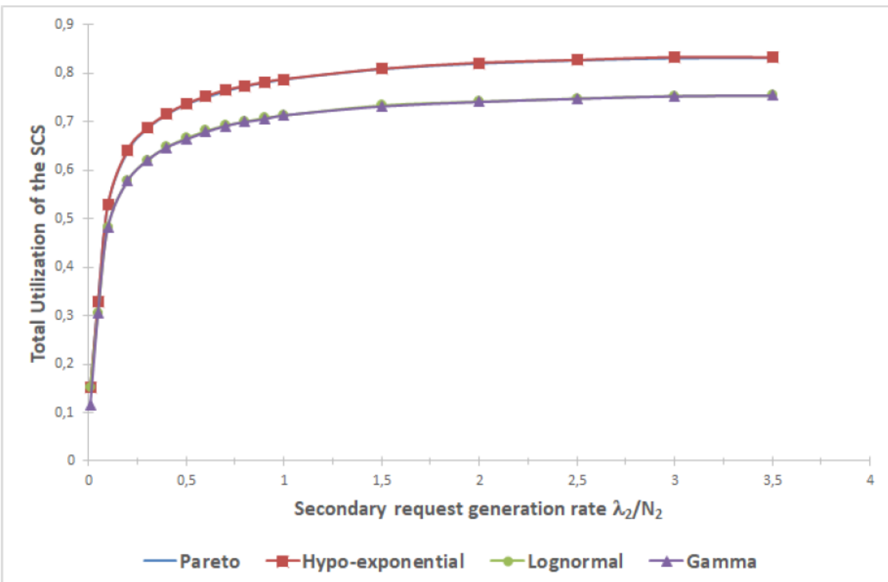


Figure 4.6: The impact of repair time distributions on the total utilization of secondary server vs λ_2/N_2

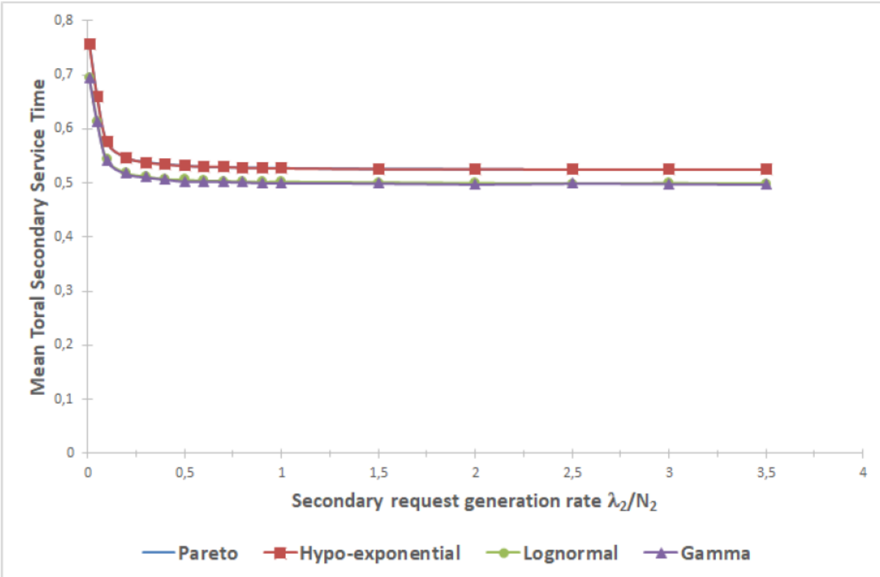


Figure 4.7: The impact of repair time distributions on the mean total secondary service time vs λ_2/N_2

4.2.3.2 Operating time is generally distributed

We are interested in the behaviour of the operating times distributions, in this section, is split into two subheadings based on the square coefficient of variation. The input parameters for the distributions of the inter-failure times are shown in Table 4.4.

Table 4.4: Parameters of the distribution

<table>
<tr><td colspan="2">Distribution</td><td>Hyper</td><td>Hypo</td><td>Gamma</td><td>Pareto</td><td>Lognormal</td></tr>
<tr><td rowspan="3">Fig. 11-13, 17-19</td><td>Mean</td><td>N/A</td><td>10</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>Var.</td><td>N/A</td><td>0.68</td><td>0.68</td><td>0.68</td><td>0.68</td></tr>
<tr><td>Par.</td><td>N/A</td><td>λ_1=0.125
λ_2=0.5</td><td>α=1.470
β=0.147</td><td>α=2.751
K=0.611</td><td>m=0.720
σ=2.043</td></tr>
<tr><td rowspan="3">Fig. 8-10, 14-16</td><td>Mean</td><td>10</td><td>N/A</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>Var.</td><td>2.56</td><td>N/A</td><td>2.56</td><td>2.56</td><td>2.56</td></tr>
<tr><td>Par.</td><td>λ_1=0.132
λ_2=0.133
p=0.330</td><td>N/A</td><td>α=0.390
β=0.0390</td><td>α=2.189
K=5.432</td><td>m=1.667
σ=1.126</td></tr>
</table>

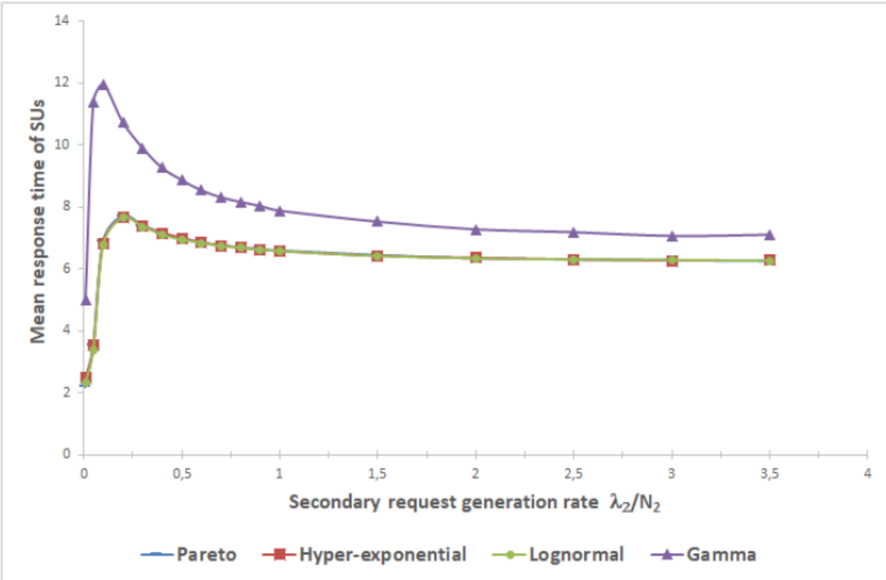


Figure 4.8: The effect of the operating time distributions on the mean sojourn time of cognitive users vs secondary request generation rate

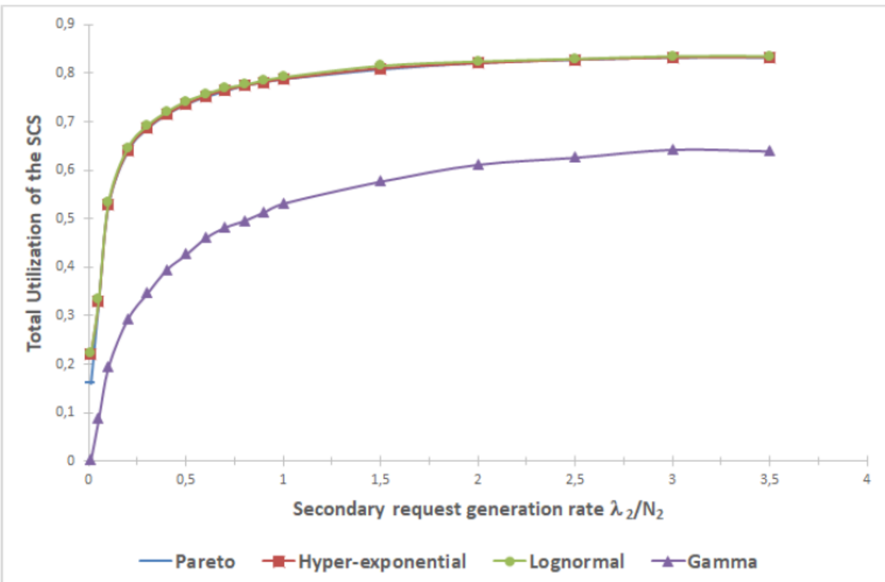


Figure 4.9: The effect of the operating time distributions on the utilization of SCS vs secondary arrival rate

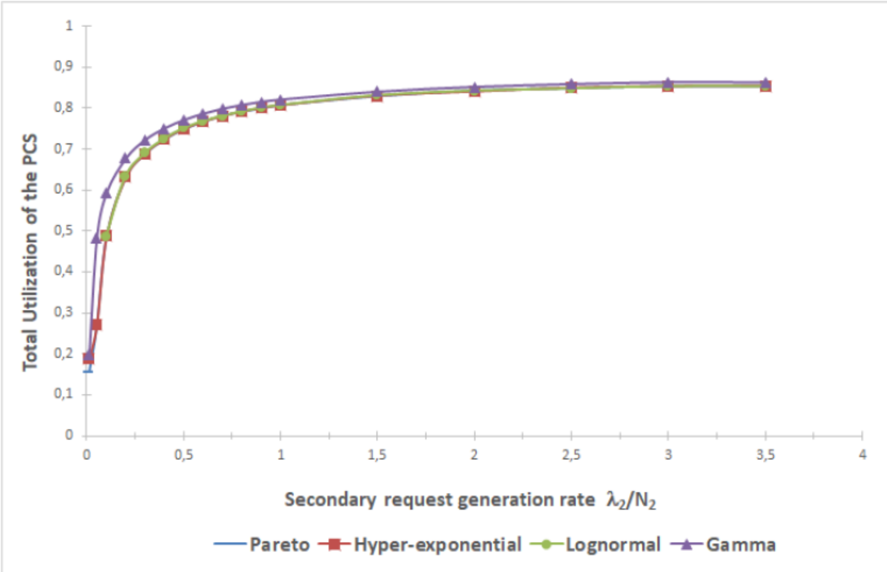


Figure 4.10: The impact of the operating time distributions on the utilization of PCS vs secondary arrival rate

Server fails during idle state Let's assume that all other random variables are exponentially distributed and that the inter-failure time of the server in an idle state is generally distributed.

Figures 4.8, 4.9 and 4.10 show how the failure time distribution impacts the mean sojourn time, and secondary and primary service units utilization, respectively, having λ_2/N_2 as the running parameter. All the distributions shown in these figures were functioning based on C_x^2 greater than 1.

The other distributions have no impact on the performance notwithstanding the gamma distribution, which shows a significant sensitivity in the estimation of the mean sojourn time and the usage. 4.10 illustrates how the gamma distribution affects the use of the main channel as well. The server is more often unavailable as a result of the gamma distribution-generated inter-failure times, and this has an effect regardless of how frequently the main service channel is used.

In order to examine the case of the involved distributions with $C_x^2 < 1$, we substituted the hyper-exponential distribution in Figures 4.11, 4.12 and 4.13 with the hypo-exponential distribution, which illustrates the same characteristics as those shown in the figures above in this section. In comparison to the equivalent figures above, the relative difference between the gamma distribution and the other distributions is relatively tiny for this set of values. The mean

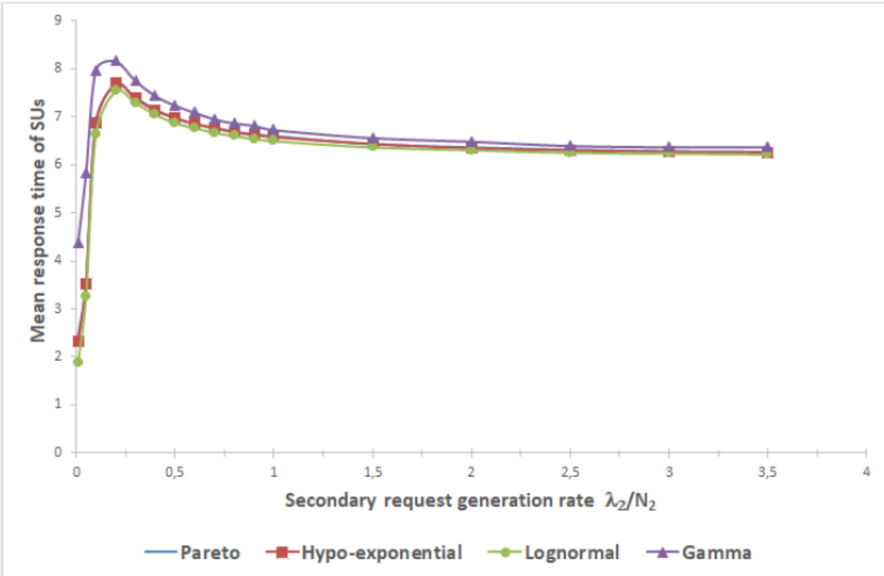


Figure 4.11: The effect of the operating time distributions on the mean sojourn time of cognitive users vs λ_2/N_2

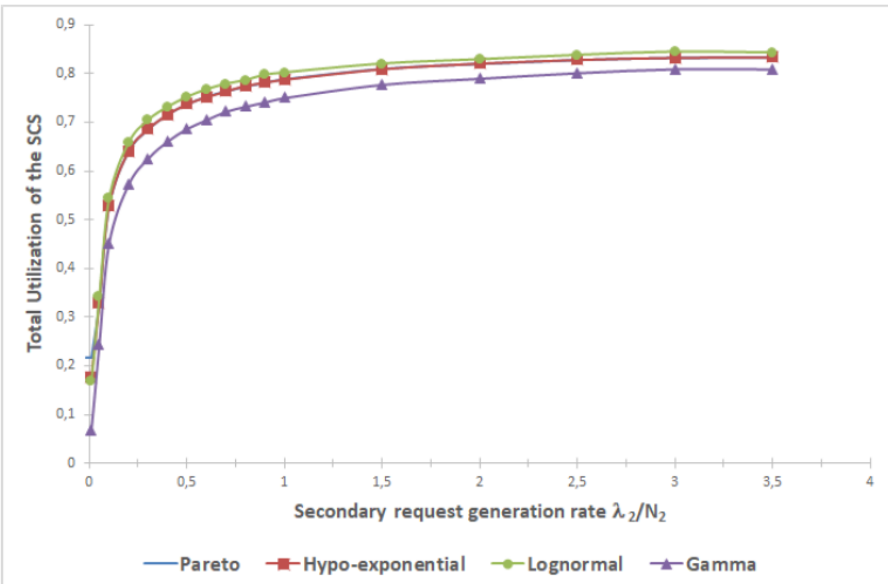


Figure 4.12: The impact of the operating time distributions on the utilization of SCS vs secondary arrival rate

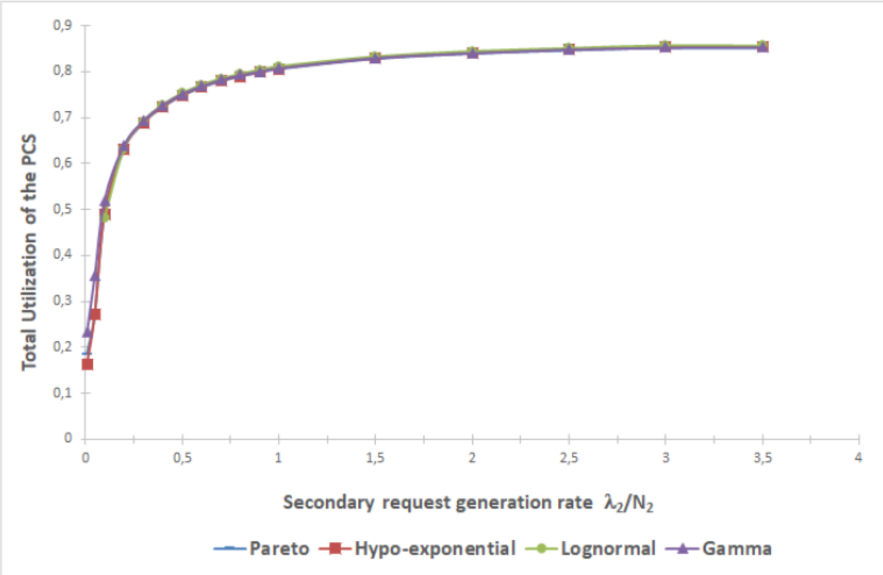


Figure 4.13: The effect of the failure time distributions on the utilization of primary server vs secondary arrival rate

residence time and SCS utilization are still somewhat impacted, but the PCS utilization is almost unaffected.

Server fails during busy state In the last investigational scenario, we assume that all other random variables are exponentially distributed and that the server failure time occurs during a busy condition. We start with the hyper-exponential distribution, the same as the prior analysis and set gamma, Pareto, and log-normal with $C_x^2 > 1$.

The inter-failure time distribution in a busy condition is shown in Figures 4.14, 4.15 and 4.16 in relation to the mean sojourn time, SCS and PCS utilization, respectively, against λ_2/N_2. Figure 4.14 depicts the effects of the distribution where the log-normal delivers the lowest value and the gamma offers the highest value of the mean response. As predicted, the gamma distribution in Figure 4.15 shows the server with the lowest use.

Instead of the hyper-exponential distribution, in Figures 4.17, 4.18, and 4.19 we included the hypo-exponential distribution. The values represent, respectively, the same estimations shown in Figures 4.14, 4.15 and 4.16.

In this instance, the inter-failure time during the idle state was generally distributed, which is the reverse of scenario 2's behaviour. In this situation,

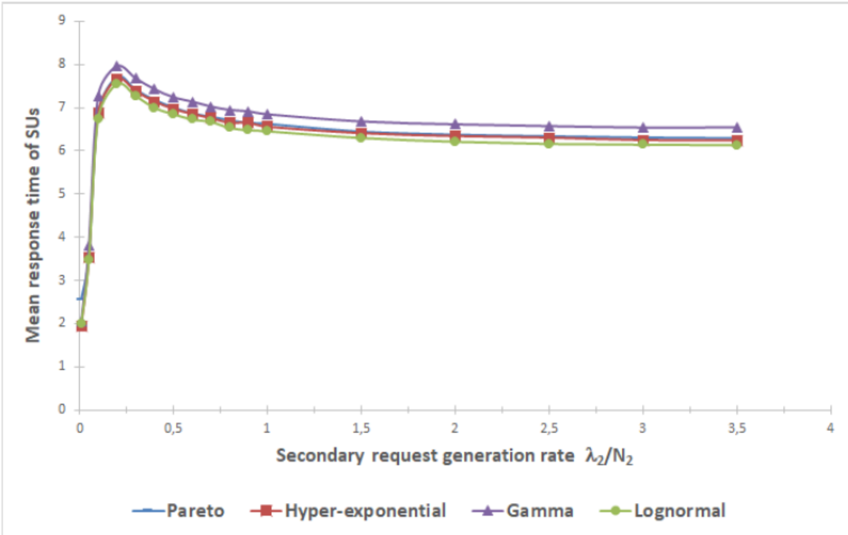


Figure 4.14: The effect of the operating time distributions on the mean sojourn time of cognitive users vs λ_2/N_2

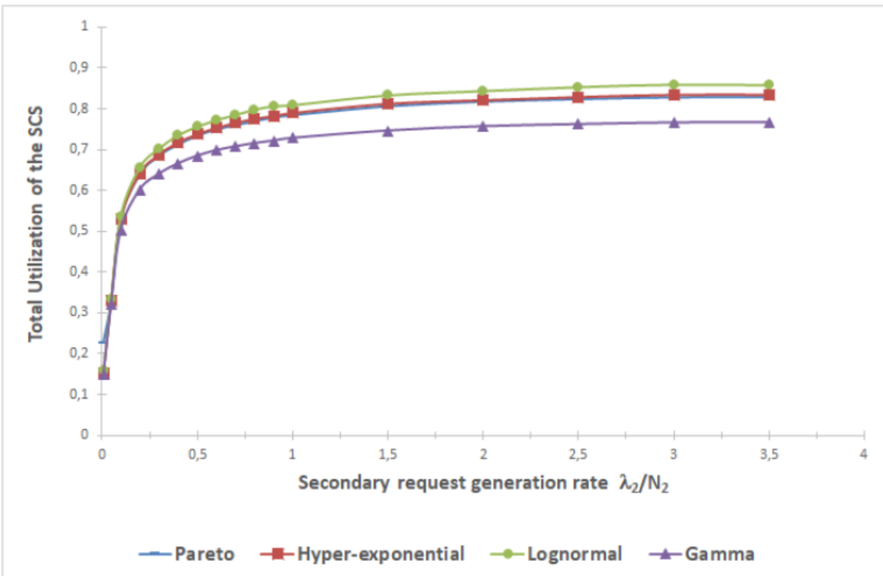


Figure 4.15: The impact of the operating time distributions on the utilization of SCS vs secondary arrival rate

the hypo-exponential distribution resulted in a lower relative difference in the estimates induced by the gamma distribution. The relative difference between

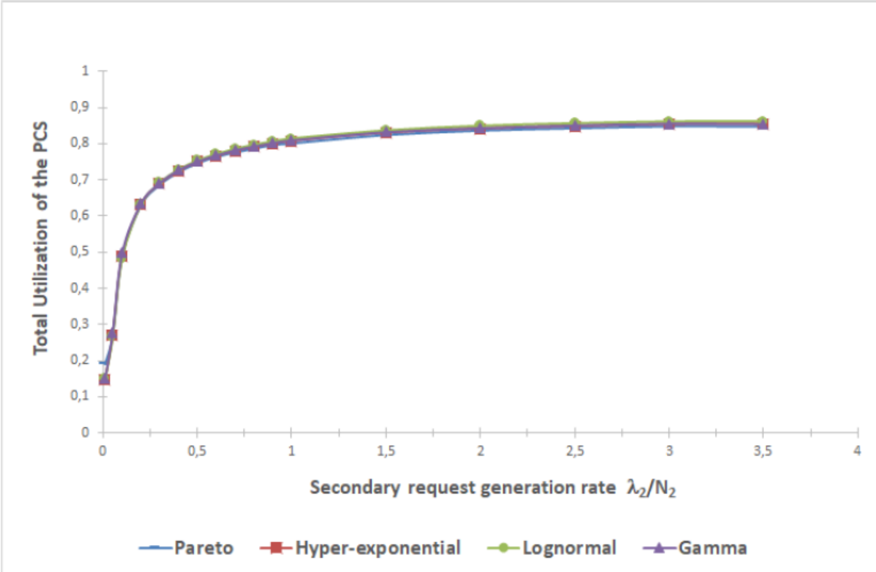


Figure 4.16: The effect of the failure time distributions on the utilization of primary server vs secondary arrival rate

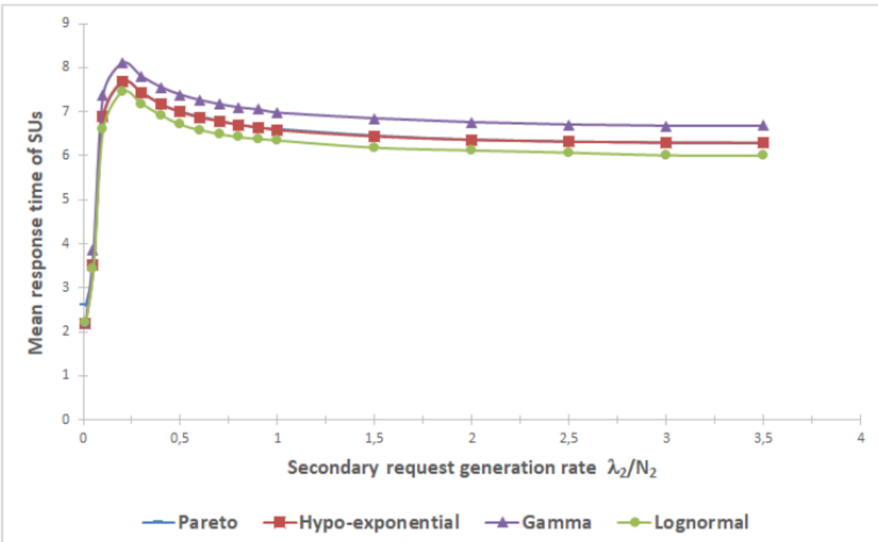


Figure 4.17: The effect of the operating time distributions on the mean sojourn time of cognitive users vs λ_2/N_2

the estimations, as shown by the figures in this scenario, is larger for the hypo-exponential distribution and less for the hyper-exponential distribution.

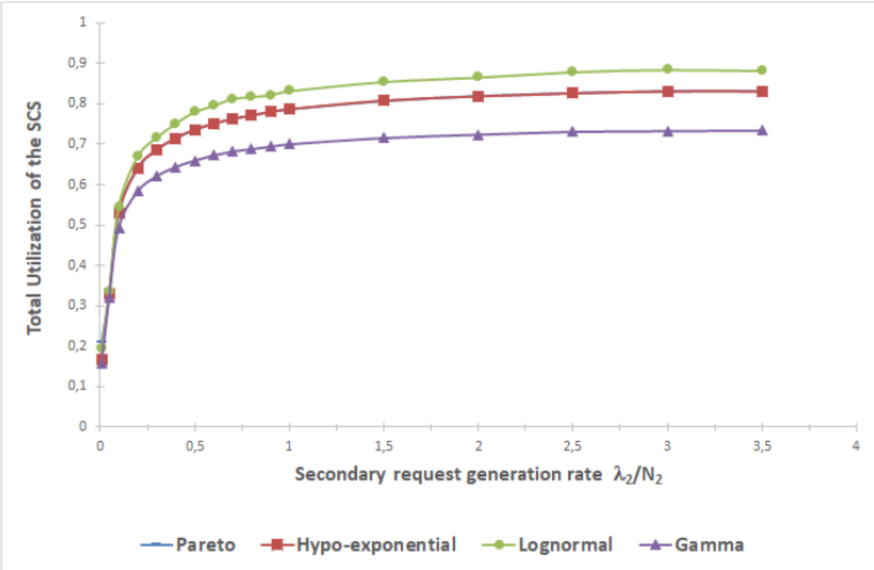


Figure 4.18: The impact of the operating time distributions on the utilization of SCS vs secondary arrival rate

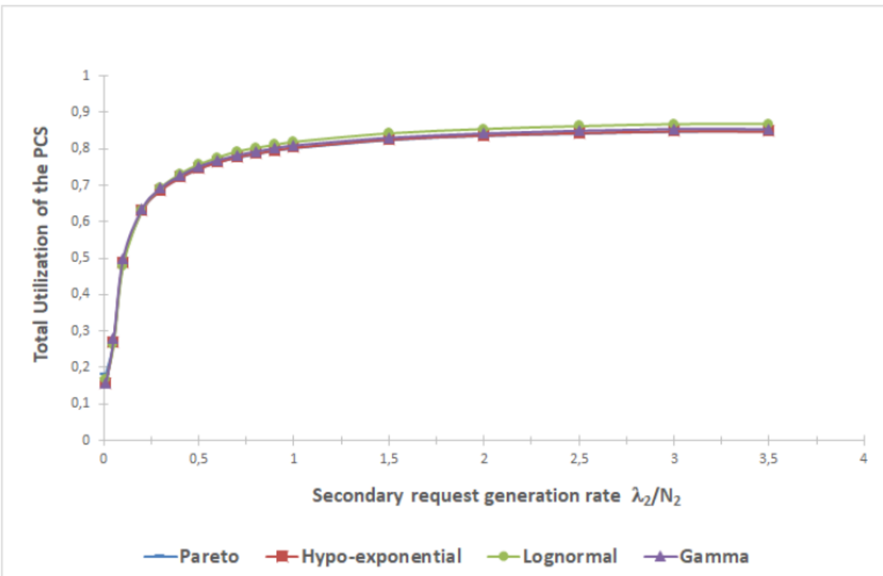


Figure 4.19: The effect of the failure time distributions on the utilization of primary server vs secondary arrival rate

Figures 4.16 and 4.19 show that the distribution of the inter-failure time during the busy state has no impact on the main service channel regardless of whether the squared coefficient of variation is greater or less than one.

However, the following is our explanation for the statistically significant difference that is depicted in Figures 4.8 and 4.14: let's take a look at these sample examples, which show respectively, the impact of the operating time distribution on the SU mean response time while the server failures occur in the idle and busy states. When the arrival rate is between [0.1 and 1], the difference becomes substantial. At this low arrival intensity, the server is more prone to breakdowns when it is in an idle state, preventing arriving users from joining the service unit and resulting in a longer response time. Due to the lowest values that the gamma distribution creates, the difference may be noticed when the operating time is gamma distributed (Short operation time).

4.3 Finite-source cognitive radio network with unreliability and abandonment

The system model from the previous section, which addressed system unreliability, is supplemented by the abandonment behaviour of SUs, this describes this section.

Once their cumulative waiting time exceeds a predetermined random maximum waiting time, unlicensed users (impatient ones) are required to exit the system. The system's secondary service unit is prone to sporadic failures and repairs. The novelty of this study is in its examination of the effects of secondary server unreliability and abandonment on several network performance measures, including the probability of abandonment, mean sojourn time of users, etc.

> **New results in (J4, C3)** *The abandonment time was generally distributed random variables with the same mean and different variance, this allows us to investigate the performance measures of a finite-source cognitive radio network with unreliability and abandonment. Investigating how these distributions affect the response time's mean was the goal. The service time distribution has a significant impact on the performance indicators, according to the findings. This impact depends on the distribution's squared coefficient of variation, mainly.*

4.3.1 System model

A finite-source queuing system that simulates the under consideration cognitive radio network is shown in 4.20. Our queuing system comprises two interconnected, non-independent subsystems. Primary requests are given the first one, where N_1 is the total number of sources. These sources will be in charge of producing high priority requests with exponentially inter-request times using the λ_1 parameter. All of the generated requests are sent to a single PCS server that is linked to a pre-emptive FIFO queue. The PU jobs' service times should also be distributed exponentially with rate μ_1.

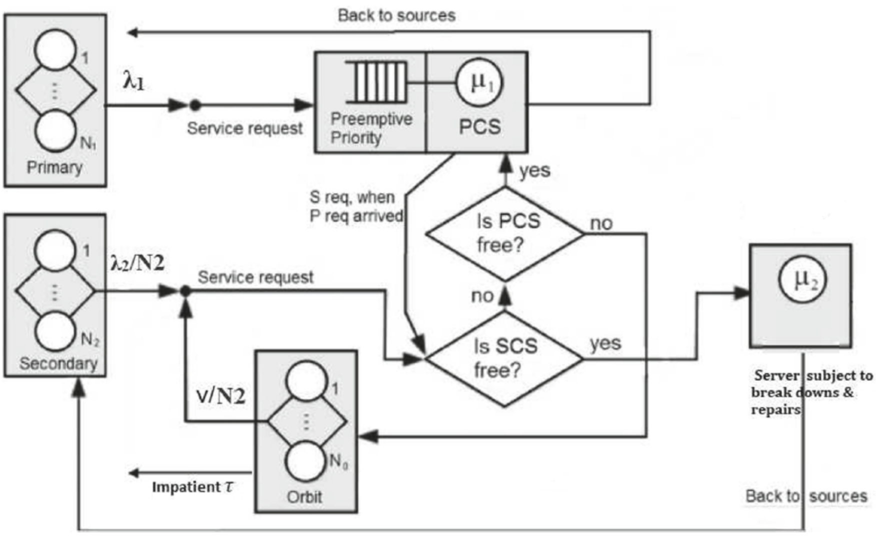


Figure 4.20: Finite-source retrial queuing system: Modeling Cognitive Radio Network with unreliability and abandonment

Low priority requests are handled by the second subsystem. Inter-arrival times and service times in this subsystem are assumed to be exponentially distributed as well with parameters N/λ_2 and μ_2, respectively. N_2 stands for the number of sources. There are two possible server states: idle and busy. Depending on the server's status, the produced main packet either enters the FIFO queue (if a PU in the PCS) or joins the primary server. If A secondary request is being handled in the PCS, that should be interrupted and instead routed back to the secondary unit, which allows PUs to start service in their dedicated unit.

The aborted task is assigned to either the secondary server or the retrial queue (orbit), depending on the availability of the secondary unit, and reties service from the beginning after an exponentially distributed period of time with a parameter ν/N_2.

On the other side, SCS receives requests from SUs. The service starts if it is idle; else, this unlicensed process will sense the PCS, service may opportunistically start in the high priority channel if PCS is in an idle state. The request is sent to the orbit if the PCS is occupied.

It should be emphasized that Secondary Users in orbit are obliged to quit the system whenever their cumulative waiting time exceeds a generally distributed random time (abandonment time), using the following distributions: Hyper, Hypo, Gamma, Log-normal and Pareto) with a rate τ. The secondary service unit may have random failures in both its busy and idle states after an exponentially distributed random period with parameters γ_1 and γ_2, respectively. Repair time is an exponential random variable, as well, having σ as a parameter.

These following notations are introduced to establish a stochastic process that describes the behaviour of the system:

- $k_1(t)$ at time t, how many primary sources we are dealing with in the system,
- $k_2(t)$ at time t, how many secondary sources we are dealing with in the system,
- $q(t)$ at time t, the number of high priority jobs in the FIFO queue,
- $o(t)$ at time t, the number of SUs in the orbit,
- $y(t) = 0$ if the PCS is empty, $y(t) = 1$ in case the PCS is occupied with a high priority job, at time t and $y(t) = 2$ in case the PCS is occupied with a low priority job,
- $c(t) = 0$ if the SCS is idle and $c(t) = 1$ if the SCS is occupied at time t.

Consequently, we can assume

$$k_1(n) = \begin{cases} N_1 - q(t), & y(t) = 0{,}2 \\ N_1 - q(t) - 1 & y(t) = 1 \end{cases}$$

$$k_2(n) = \begin{cases} N_2 - o(t) - c(t), & y(t) = 0{,}1 \\ N_2 - o(t) - c(t) - 1 & y(t) = 2 \end{cases}$$

Table 4.5: Confidence intervals of the figures

Fig.	Obs. Point	Distubition	N	$t_{N,1-\frac{\beta}{2}}$	95% Confidence Int.	
					LB	UB
4.23	2	Hyper	69	1.995	97.012	104.789
		Gamma	72	1.994	96.892	104.516
		Lognormal	65	1.997	97.152	105.016
		Pareto	68	2.000	98.002	104.891
4.24	1.8	Hypo	86	1.988	3.023	3.595
		Gamma	81	1.988	3.812	6.492
		Lognormal	79	1.990	3.141	3.749
		Pareto	89	1.987	3.589	4.202

4.3.2 Validation of results

We developed a stochastic simulation program implemented in the C programming language using SimPack [19] modules under the assumption that all random variables included in the system are exponentially distributed, except abandonment time, which is a generally distributed random variable.

The key steady-state performance measurements may be achieved by constructing a continuous-time Markov chain in the case of exponentially distributed inter-event times, as was done in [2]. The numerical results in this instance were test results that the C language validated, based on SimPack [19].

We are involving more general situations that allow non-exponentially distributed random variables, nevertheless. The values of the input parameters are gathered in Table 4.6, for ease of comprehension.

The confidence intervals values of some sample figures are shown in Table 4.5

Table 4.6: Values of simulation parameters

N_1	N_2	λ_1	λ_2/N_2	μ_1	μ_2	ν/N_2	τ	γ_1	γ_2	σ
10	100	0.01	x-axis	4	4	0.01	0.002	0.01	0.01	1

4.3.3 Numerical results

Using simulation, multiple scenarios are examined in this section, this allowed us to compare different observations within a single run.

Table 4.7: Different impatience rates

	N_2	λ_2/N_2	ν/N_2	γ_1	γ_2	σ	μ_1, μ_2	τ
Case 1	100	0.01	0.1	0.1	0.1	1	1	0.000001
Case 2	100	0.01	0.1	0.1	0.1	1	1	0.0001
Case 3	100	0.01	0.1	0.1	0.1	1	1	0.001
Case 4	100	0.01	0.1	0.1	0.1	1	1	0.01
Case 5	100	0.01	0.1	0.1	0.1	1	1	0.1
Case 6	100	0.01	0.1	0.1	0.1	1	1	1

We were able to compare the performance of two categories of cognitive customers (SUs), those who leave the system after receiving successful service and those who abandoned the system without receiving a service as a result of waiting time exceed. Additionally, there is a distinction between secondary users who successfully received service from the primary service channel and secondary users who leave the system without receiving service following numerous interruptions at the primary service unit, caused by primary customers' priority over them. In the simulation, we employed the batch mean approach to estimate the performance metrics for these categories.

- **Scenario 1**: The users' impatience time is exponentially distributed.
- **Scenario 2**: Impatience time is generally distributed using the Hyper-Exponential, Gamma, Lognormal, and Pareto distributions, with $C_x^2 > 1$.
- **Scenario 3**: impatience time is generally distributed using the Hypo-Exponential, Gamma, Lognormal, and Pareto distributions with $C_x^2 < 1$.

In the aforementioned circumstances, it is assumed that the secondary service interruption brought on by the delivery of PUs or the server failure in the SCS would be repeated from the beginning (non-intelligent). Additionally, even if a service unit fails, the system won't be blocked, and free sources continue to generate new calls.

4.3.3.1 Impatience time is exponentially distributed

Assuming that all the random inter-times are exponentially distributed random variables in this section, we tried to analyse the key aspects of the system by focusing on increasing τ.

The set of parameter values defined in Table 4.7 were used to obtain the following results of this section.

Table 4.8: Estimation of the expectations for scenario A

	$E(T\ S)$	$E(W\ S)$	$E(T)$	$E(W)$	$E(N\ S)$	$E(T\ A)$	P_a
Case 1	14.0437	13.8001	14.04	13.8001	48.59	0.0000	0.0000
Case 2	14.0525	13.8165	14.05	13.8284	44.64	15.0001	0.001
Case 3	13.8333	13.5979	13.59	13.0235	38.17	15.226	0.012
Case 4	12.3461	12.1107	12.48	12.27	28.33	13.5472	0.15
Case 5	5.5598	5.3241	6.0853	5.9801	12.21	6.4914	0.56
Case 6	0.8258	0.5908	0.9654	0.3491	0.9772	0.9772	0.9217

Table 4.9: Estimation of the variances for scenario A

	$Var(T\ S)$	$Var(W\ S)$	$Var(T)$	$Var(W)$	$Var(T\ A)$
Case 1	197.227	190.66	197.227	197.227	0.0000
Case 2	197.473	190.897	197.47	190.87	185.249
Case 3	191.36	184.902	191.35	181.35	185.652
Case 4	152.42	146.67	152.48	146.66	183.52
Case 5	30.9119	28.3471	30.91	28.43	42.13
Case 6	0.6818	0.3491	0.6820	0.3491	0.955

Thanks to the simulation, we present precise estimations for the two forms of cognitive users, those who were successfully serviced and those who were abandoned, in the sections below. Tables 4.8, 4.9, 4.10 and 4.11 provide certain system characteristics for which notations are given in Table 4.12. The observations imply the measurements' estimated means and variances based on two scenarios:

- **Scenario A**: There are just a few sources $N_1 = 10$ and a low volume of primary customers arriving $lambda_1 = 0.01$.
- **Scenario B**: We deal with a significantly greater number of sources $N_1 = 100$ and a heavier arriving traffic $lambda_1 = 0.1$.

The values of the variances and expectations for the several categories of cognitive users are shown in Tables 4.9 and 4.8, respectively. Our results are obtained in this section with $\lambda_1 = 0.01$ and $N_1 = 10$. The instances when the impatience rate τ is increasing are identified in the rows of the tables. It is obvious that when the probability of abandonment is increasing (along with the impatience rate), the mean and variance values of the response and waiting times for arbitrary users are reduced.

Unexpectedly, we were able to observe the instability of the variance and mean values of the impatient users, while the abandonment rate was increasing. This occurrence could be explained as follows:

Table 4.10: Estimation of the expectations for scenario B

	E(T S)	*E(W S)*	*E(T)*	*E(W)*	*E(N S)*	*E(T A)*	P_a
Case 1	25.0657	24.8343	25.0651	24.8338	57.28	29.5803	0.000023
Case 2	24.9055	24.66	24.8423	24.6119	57.28	26.5792	0.002
Case 3	24.2940	24.0632	24.3561	24.1307	52.26	26.9554	0.02
Case 4	20.0967	19.8643	20.4913	20.3042	30.15	22.118	0.194
Case 5	6.5178	6.2909	7.2363	7.1664	6.07	7.5563	0.61
Case 6	0.8242	0.6068	0.9743	0.3682	0.9598	0.9834	0.943

Table 4.11: Estimation of the variances for scenario B

	Var(T S)	*Var(W S)*	*Var(T)*	*Var(W)*	*Var(T A)*
Case 1	628.037	616.75	628.29	616.57	726.24
Case 2	620.037	608.519	608.59	606.84	724.40
Case 3	590.201	579.04	590.2013	579.03	726.65
Case 4	403.8815	394.591	403.8813	394.5913	489.2139
Case 5	42.4828	39.5766	0.6794	39.4741	57.0989
Case 6	0.6771	0.3682	0.6794	0.3682	0.9671

Customers seldom exit the system when the impatience rate is extremely low as their wait times are so high, and as a result, the estimation is inaccurate since the confidence interval of the expectation for a small collection of observations might be excessively wide.

The same aspects in Tables 4.10 and 4.11 were addressed as the above section, however, in this instance, the primary traffic was more intensive considering N_1 was set to 100 and $\lambda_1 = 0.01$. The efficiency of cognitive technology is the first thing we notice when comparing it to Tables 4.8 and 4.9.

In scenario A, at a very low impatience rate ($\tau = 0.000001$), the probability of abandonment is 0 in Table 4.8, row 1, due to the lack of primary users at the licensed service channel.

Despite the larger mean and variance values in scenario B, we perform the same analysis to explain the expectation and variance of waiting time of the impatient users (Table 4.12).

4.3.3.2 Impatience Time is Generally Distributed with $C_x^2 > 1$

This section covers the second investigational scenario, including (hyper-exponentially, gamma, Pareto, and lognormally) distributions with the same

Table 4.12: Expectations and variances notations

Notation	Definition
E(T S), Var(T S)	Mean and variance response time of successful cognitive users
E(W S), Var(W S)	Mean and variance wating time of successful cognitive users
E(T), Var(T)	Mean and variance response time of arbitrary cognitive users
E(W), Var(W)	Mean and variance waiting time of arbitrary cognitive users
E(N S)	Mean number of secondary customers in the system
E(T A), Var(T A)	Mean and variance waiting time of impatient customers
P_a	Probability of abandonment

Table 4.13: Parameters of the distributions

Distrubtions	Gamma	*Hyper-exponential*	*Pareto*	*Lognormal*
Parameters	$\alpha = 0.390625$ $\beta = 0.0007813$	$p = 0.33098$ $\lambda_1 = 0.00132$ $\lambda_2 = 0.00268$	$\alpha = 2.1892$ $k = 270.5630$	$m = 5.5797$ $\sigma = 1.12684$
Mean	500			
Variance	640000			
C_x^2	2.56			

mean and variances, in order to look into the impact of abandonment time distributions on the key system characteristics. See [36] for information on determining the size, shape, and rate of these distributions. The distribution parameters' numerical values are defined in Table 4.13.

A variety of random number generator techniques are used to create the inter-event times. The input parameters for these methods, in our instance the rates of the distributions, are necessary. Table 4.6 defines these parameters' numerical values.

Comments It should be noted that in all of the obtained figures, apart from the abandonment time, all of the inter-event times in the system are assumed to be random variables exponentially distributed.

Figure 4.21 demonstrates how the abandonment time distributions affect the mean sojourn time of cognitive users leaving the system following a successful service, as the secondary arrival intensity rises. The lowest mean response value is produced by the Pareto distribution, whilst the largest value is produced by the gamma distribution. The density function of each distribution explains this relative disparity. Additionally, if we review articles that examined a single server finite-source retrial queuing system with abandonment such as [49], we notice that the relative difference between the mean values of the lognormal and hyper-exponential distributions is less than that found in 4.20. This distinction

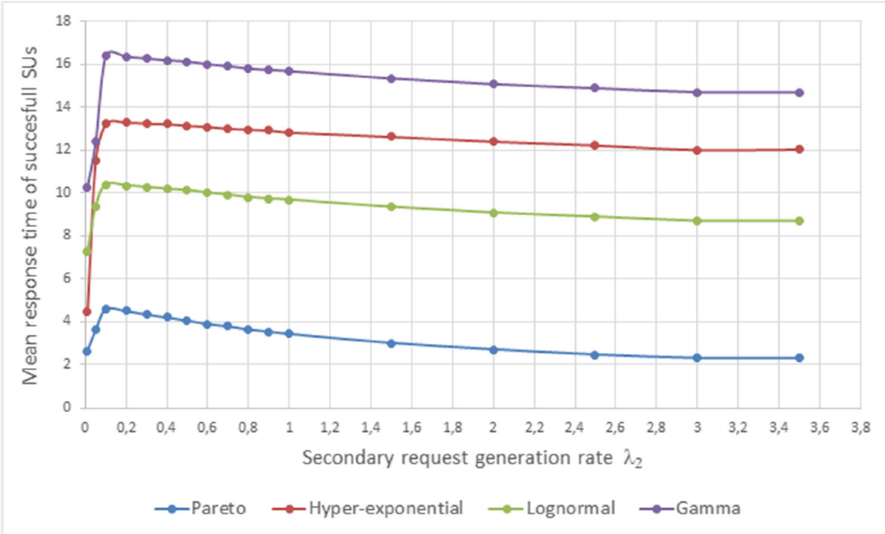


Figure 4.21: The impact of the impatience time distributions on the mean sojourn time of successful cognitive users vs secondary request generation rate

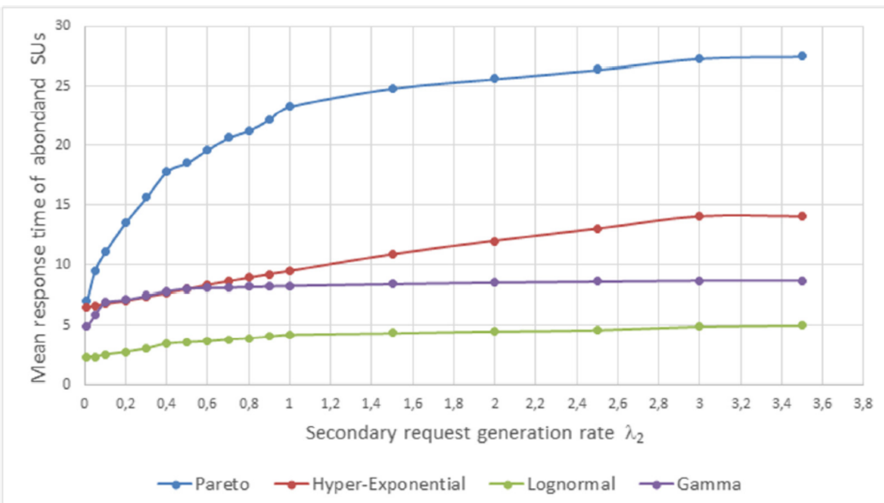


Figure 4.22: The impact of the impatience time distributions on the mean sojourn time of impatient cognitive users vs secondary request generation rate

is clearly the result of adding a second server and giving secondary users a cognitive type of behavior, both of which increase system complexity. However, we were able as well to reach the maximal property of the mean response time that was reported in [42].

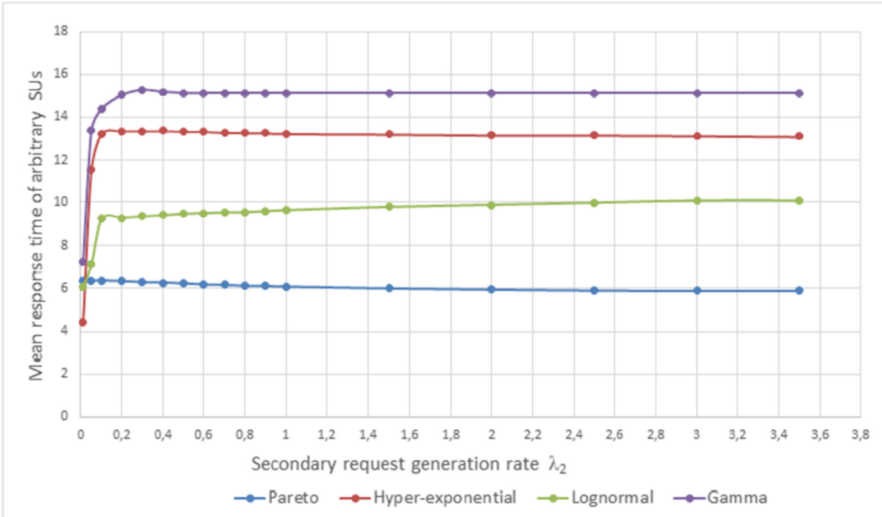


Figure 4.23: The impact of the impatience time distributions on the mean sojourn time of arbitrary cognitive users vs secondary request generation rate

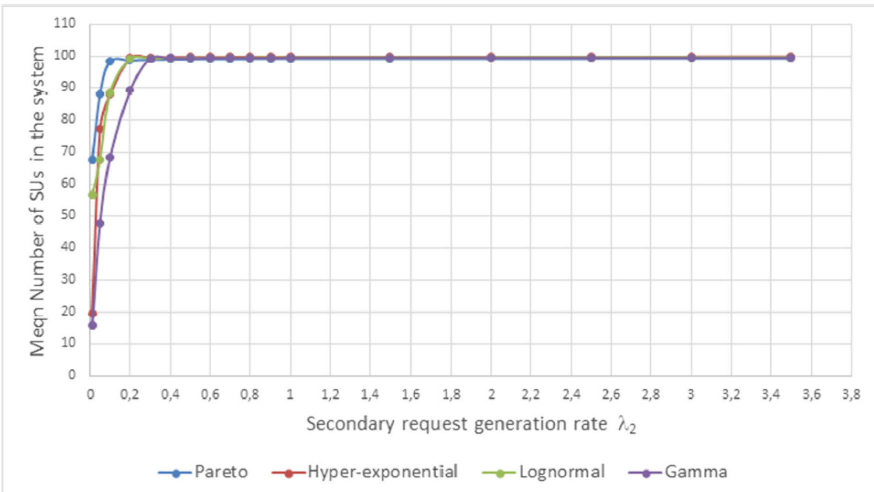


Figure 4.24: The mean number of cognitive users in the system vs secondary arrival intensity

The impact of the impatience time distributions on the mean sojourn time of impatient SUs is shown in Figure 4.22. The result demonstrates that the mean value rises as the intensity of request generation increases. The Pareto distribution delivers the biggest mean value, while the log-normal distribution displays the least value for this feature, achieving the goal of distinguishing

Table 4.14: Parameters of the distributions

Distrubition	Gamma	Hypo-exponential	Pareto	Lognormal
Parameters	$\alpha = 1.47059$ $\beta = 0.002941$	$\lambda_1 = 0.01$ $\lambda_2 = 0.0025$	$\alpha = 2.5718$ $k = 305.\ 5844$	$m = 5.9552$ $\sigma = 0.72027$
Mean	500			
Variance	170000			
C_x^2	0.68			

between successful and impatient customers.

It is simple to determine the mean of arbitrary users using the law of total expectation. Figure 4.23 illustrates how the impatience time distribution affects the mean response time of an arbitrary cognitive user as the rate of secondary request generation rises. The probability of abandonment and success affect the mean value of arbitrary users. Due to the high probability of abandonment associated with a small value of impatient time provided by the gamma distribution, the value of the mean response time of a randomly selected user is the highest, as seen in this figure.

In Figure 4.24, we were interested in the mean number of secondary customers in the function of the second generation request rate while the impatience time is generally distributed. When the system is not heavily loaded, the effects of the distributions may be seen. The mean number of cognitive users grows as the arrival intensity increases, and the distributions have less of an effect on the value.

4.3.3.3 Impatience Time is Generally Distributed with $C_x^2 < 1$

In the third scenario of our analysis, we adjust the distribution's characteristics so that their coefficient of variation is less than one. We use the two phases of the hypo-exponential distribution in this instance, trying to find out their impact on the system's key performance measures. Table 4.14 provides the updated set of distribution parameters.

Comments Figures 4.25 and 4.26 demonstrate, in regards to arrival intensity, the mean residence times of successful and random users, respectively. The output analysis reveals the same pattern as the previous related figures, however, some variations can be observed, particularly in the gamma distribution.

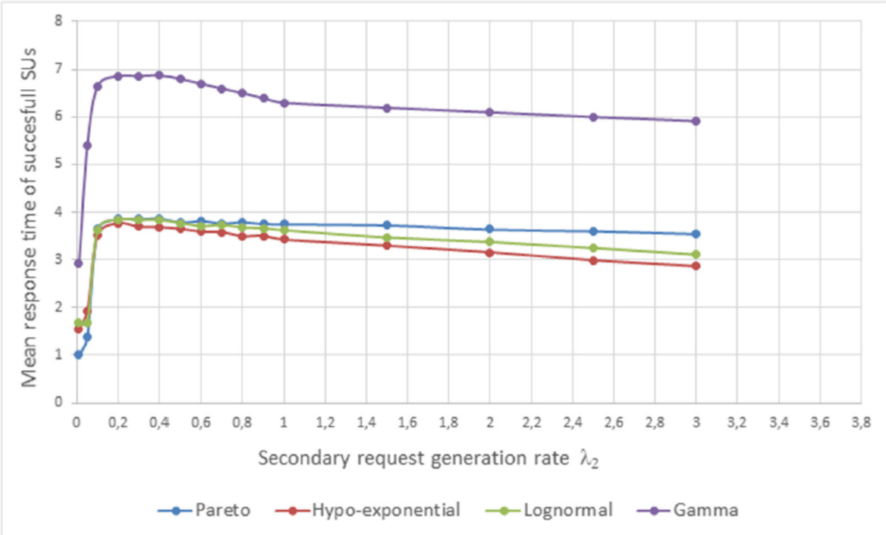


Figure 4.25: The effect of the impatience time distributions on the mean sojourn time of successful cognitive users vs secondary request generation rate

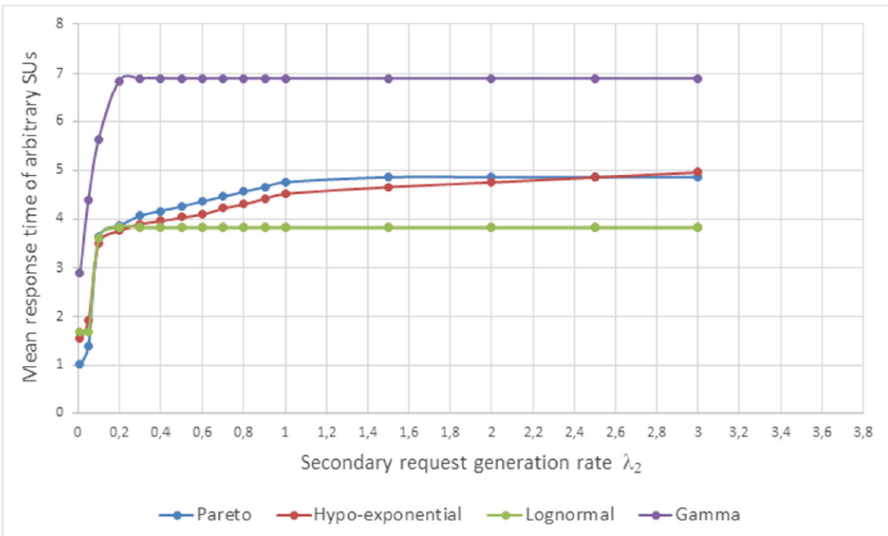


Figure 4.26: The effect of the impatience time distributions on the mean sojourn time of arbitrary cognitive users vs secondary request generation rate

Additionally, these analyses show that both, on average, successful and random users utilize the system for shorter periods.

The average number of secondary customers in the system as a function of

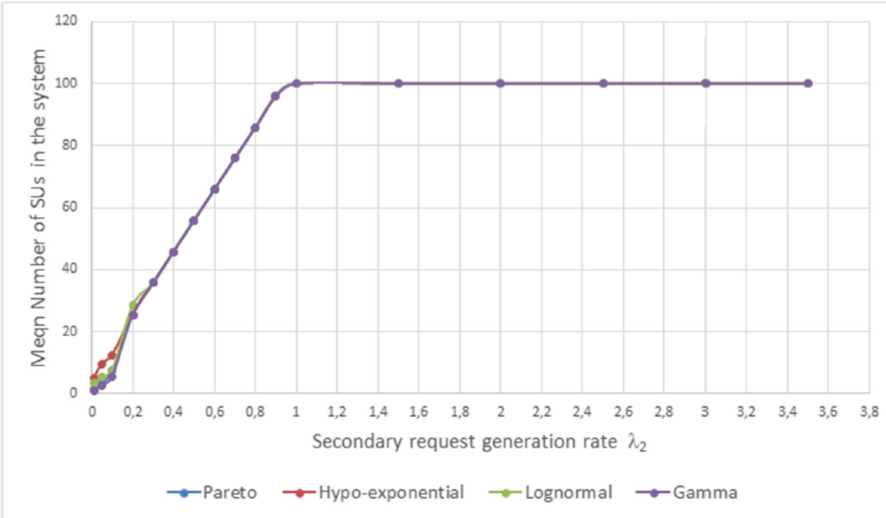


Figure 4.27: The mean number of cognitive users in the system vs secondary arrival intensity

secondary arrival intensity is shown in Figure 4.27, which demonstrates no effect of the distribution on the mean number of customers in a highly loaded system, but in a low loaded system, a slight difference can be noticed.

Additionally, when the squared coefficient of variation of the distributions is less than one, there are fewer users in the system in comparison to the prior set of parameters with the corresponding figure.

4.4 Analysis of finite-source cognitive radio network with balking and reneging

The theories of balking (refusing to join the queue) and reneging (leaving the system after joining) as they relate to cognitive radio networks are covered in this section. These decisions that users could make while waiting in a line are among the most well-known behaviours. New arriving customers are discouraged to join the system as it gets more crowded; nonetheless, impatient users will exit after entering, once their waiting time exceeds a maximum value.

Myriad investigations dealt with balikng and reneging on several queuing systems, including [58] and [35]. We were the initiators of including these features on this CRN. Assuming that the secondary service time is generally distributed, we aimed to study these distribution's effects and the impact of cognitive technology on the key performance measures of our system.

4.4.1 System model

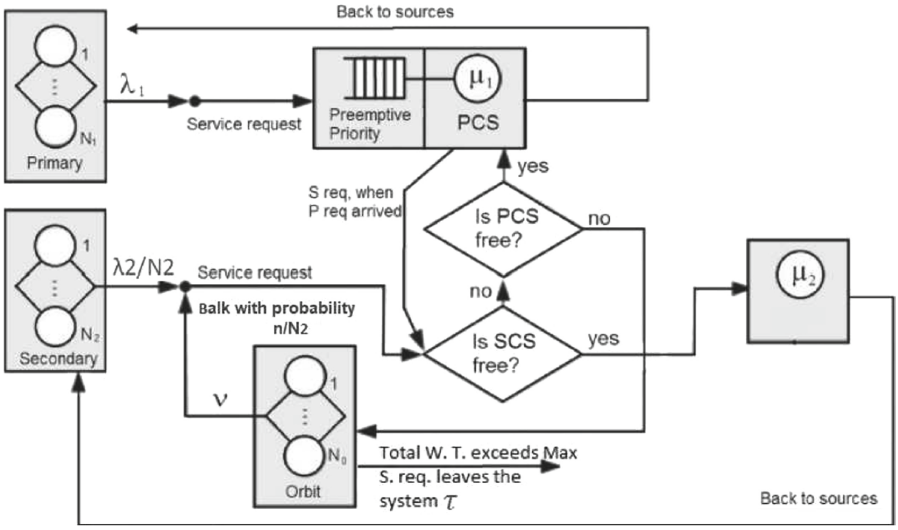


Figure 4.28: Finite-source retrial queuing system: Modeling the Cognitive Radio Network with balking and reneging

The queuing cognitive radio system based on the following assumptions is shown in Figure 4.28.

Assuming that our system has the basic features of the cognitive radio network, we consider two interconnected subsystems where the first server receives primary requests from a limited number of sources N_1 over an exponentially distributed period of time with a mean value of $1/\lambda_1$. If the unit is unoccupied, the service might begin; otherwise, the call would be placed in the preemptive priority queue. The primary customers' service time is a random variable with an exponential distribution and the parameter μ_1.

N_2 stands for the secondary subsystem's source number. Each source generates low-priority tasks according to an exponentially distributed time with the parameter λ_2/N_2.

With a rate μ_2, the service time of SUs is generally distributed using the same mean and different variances. The involved distributions are hypo-exponential, hyper-exponential, and gamma distributions. It is considered that the retrial

Table 4.15: Confidence intervals of the figures

Fig.	Obs. Point	Distubition	N	$t_{N,1-\frac{\beta}{2}}$	95% Confidence Int.	
					LB	UB
4.28	0.5	Hyper	68	1.995	4.587	5.293
		Hypo	70	1.994	4.662	5.323
		Gamma with $C_x^2 > 1$	65	1.997	3.452	2.805
		Gamma with $C_x^2 < 1$	60	2.000	6.993	7.55
4.32	0.06	Hyper	86	1.988	65.201	58.952
		Hypo	85	1.988	85.145	90.987
		Gamma with $C_x^2 > 1$	79	1.990	96.130	102.542
		Gamma with $C_x^2 < 1$	90	1.987	7.829	13.013

time for the secondary customer is exponentially distributed with a parameter ν.

With a probability of n/N_2, where n is the total number of sources and N_2 is the total number of users in the system, newly arriving secondary customers may balk (refuse to join the server). Furthermore, if the service does not begin by a certain random time, which is exponentially distributed with a parameter τ, they may also renege (leave the orbit after joining).

> **New results in (J2)** *We followed a line of investigation on the finite-source cognitive radio network presuming that cognitive users might balk and renege. We let the service time of SUs be generally distributed. The aim was to delve into studying the impact of cognitivity and service time distributions on our system's key performance measures.*

4.4.2 Validation of the results

The numerical outcomes in this chapter's section illustrate the influence of service time distributions on the key performance measures as well as the impact of the balking and reneging on the system's retrial component. Precise confidence intervals for the estimations of the means response times for chosen figures are shown in Table 4.15.

4.4.3 Numerical results

We used SimPack [19] to develop a stochastic simulation program on C language with the assumption that all random variables present in the system, except the

Table 4.16: Parameters of the general distributions

Distribution	Gamma with $C_x^2 < 1$	Hyper	Hypo	Gamma with $C_x^2 > 1$
Parameters	α=1,7857 β=1,7857	p = 0,3309 λ_1 = 0,66198 λ_2 = 1,33803	λ_1= 1,4854 λ_2= 3,06	α= 0,3906 β= 0,3906
Mean	1	1	1	1
Variane	0.56	2.56	0.56	2.56
C_x^2	0.56	2.56	0.56	2.56

Table 4.17: Simulation input parameters

N_1	N_2	λ_1	λ_2/N_2	μ_1	μ_2	ν	τ
20	50	0.1	x-axis	1	1	0.1	0.1

services, are exponentially distributed. The validation of the simulation outputs yielded all of the numerical results. While Table 4.17 lists the simulation's main class input parameters values, Table 4.16 displays the numerical values of the parameters of the distributions.

4.4.3.1 Service times are generally distributed

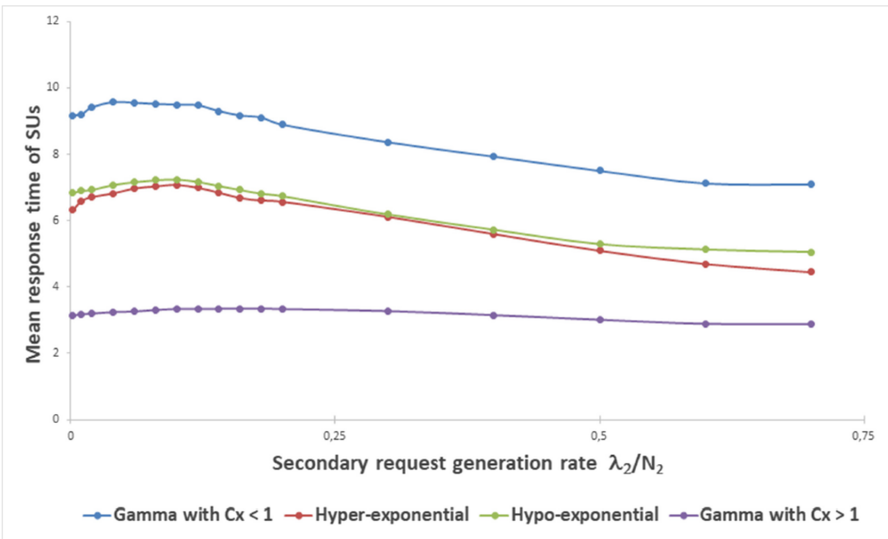


Figure 4.29: The impact of primary and secondary service times distribution on the mean residence time of SUs vs secondary request time generation

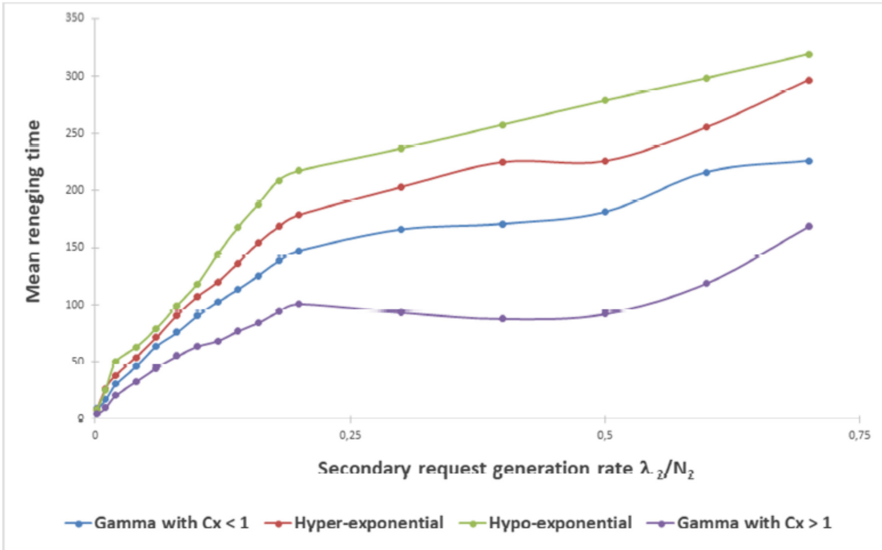


Figure 4.30: The impact of primary and secondary service times distribution on the mean reneging time of SUs vs secondary request time generation

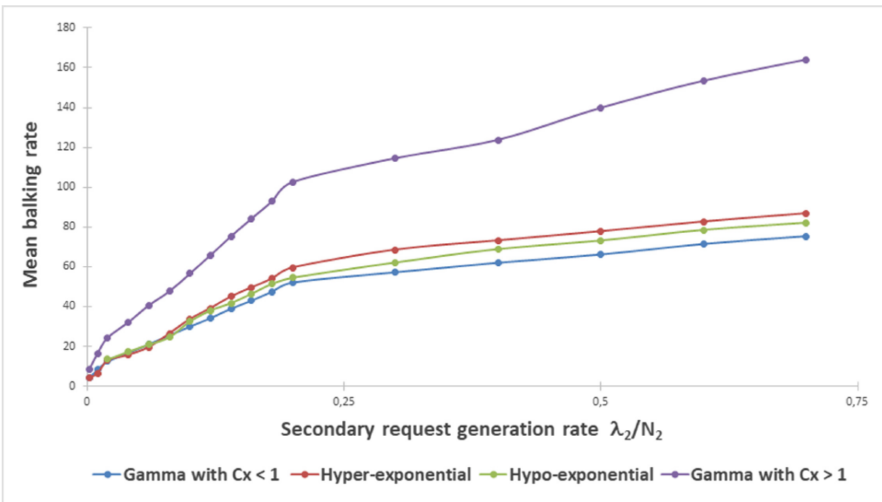


Figure 4.31: The impact of primary and secondary service times distribution on the mean balking rate of SUs vs secondary request time generation

The way the mean sojourn time of SUs is effected by the primary and secondary service time distributions VS secondary inter-arrival time is shown in Figure 4.29. When service times are gamma distributed with a $C_x^2 > 1$, a significant distribution sensitivity may be seen. The impact of primary and secondary

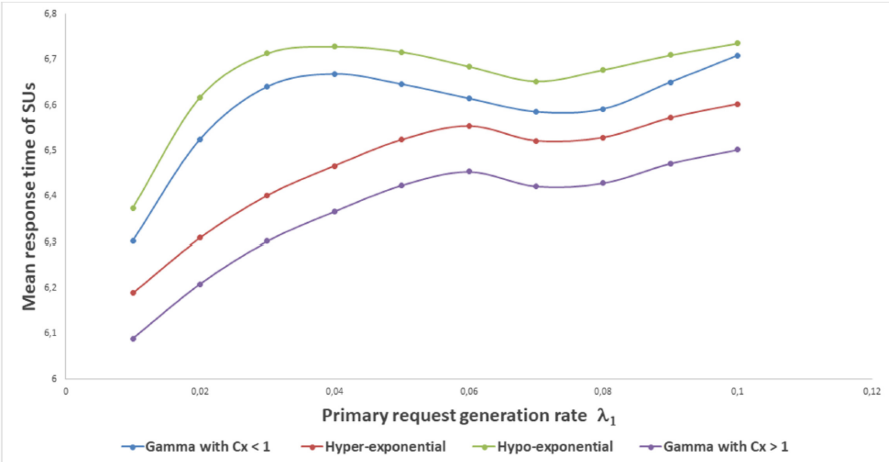


Figure 4.32: The impact of primary and secondary service times distribution on the mean residence time of SUs vs primary request time generation

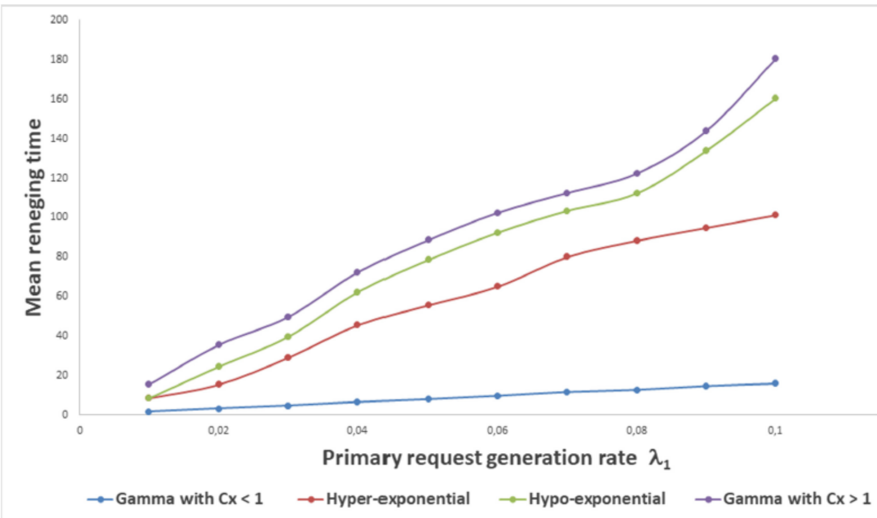


Figure 4.33: The impact of primary and secondary service times distribution on the mean reneging time of SUs vs primary request time generation

service times distribution on the mean reneging time of SUs vs secondary request time generation was illustrated in Figure 4.30, whereas gamma with a $C_x^2 > 1$ greater than one, demonstrating the same obvious sensitivity. This matches the same behavior found in earlier results. Additionally, as anticipated, a significant amount of users abandon the system as a result of the SUs' increased arrival intensity, particularly in the hypo-exponential case. The general distributions'

Table 4.18: Simulation input parameters for Figures 4.32 and 4.33

N_1	N_2	λ_1	λ_2/N_2	μ_1	μ_2	ν	τ
20	50	x-axis	0.14	1	1	0.1	0.1

parameters are shown in Table 4.16.

Figure 4.31 shows the effect of the primary and secondary subsystems' service times distribution on the average balking rate vs λ_2. Increasing the secondary arrival rate results in higher discouragement for newly arriving secondary users, this is very obvious, as shown by Gamma distribution. According to the Gamma distribution function, it is well known that when $C_x^2 > 1$, the produced random service time is high, which causes the system to become overloaded.

The effect of primary and secondary service times distribution on the average response time of SUs compared to primary request time generation is shown in Figure 4.32. This figure clearly illustrates the effect, particularly when the squared coefficient of variation is more than one.

To determine the effects of primary and secondary service times distribution on the mean SU reneging time vs primary request time generation, Figure 4.33 was generated. Employing the Gamma distribution, which has a squared coefficient of variation of less than one, did not result in a high mean reneging rate compared to the other distributions, increasing the main request generation rate, however, does result in a greater mean reneging time.

4.4.3.2 All Inter-event times are exponentially distributed

We assume that all inter-event times are exponentially distributed in this subsection. The values in Table 4.17 are applied with the same value for $\lambda_2 = 0.5$. We were interested in finding out how cognitive technology affects the system's properties.

Figure 4.34 depicts the effects of the primary inter-arrival rate and N_2 on the average response time of low-priority users. We can see from this figure that the primary arrival intensity has a significant impact on the average SU sojourn duration since the results are less if $\lambda_1 = \lambda_2/2$ than when $\lambda_1 = \lambda_2 * 2$. Contrary to N_2, which has no impact, the main sub-traffic subsystem's intensity increases when N_2 is high. However, when the primary number of sources is greater, Figure 4.35 shows a distinct effect. The reason for this is that more secondary users are leaving the system (reneging). The only influence that can be seen when the primary arrival rate is half the secondary's with only a few primary

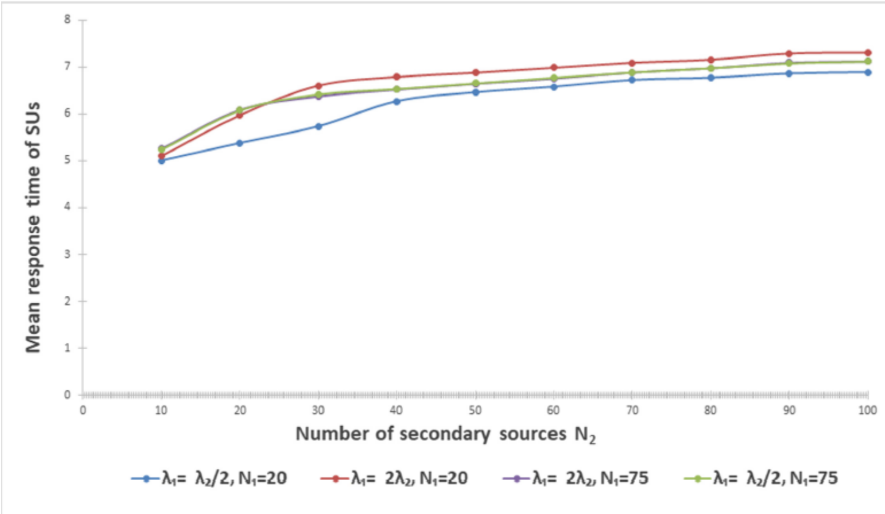


Figure 4.34: The effect of the primary network parameters on the mean response time of SUs vs N_2

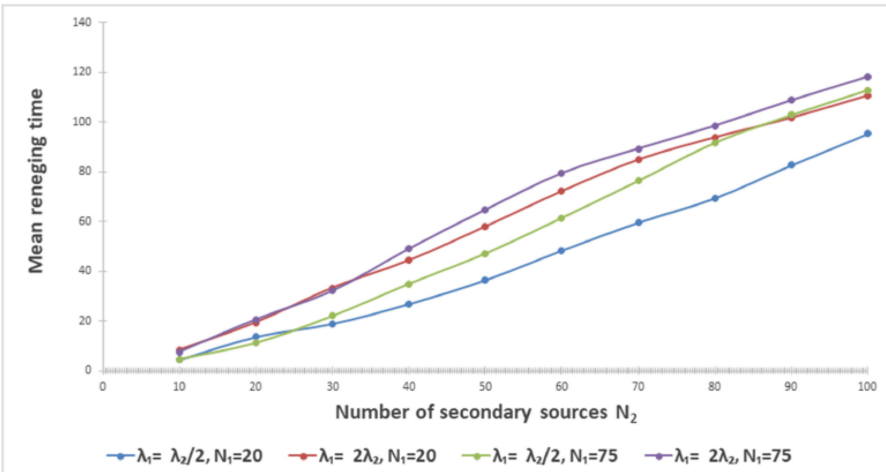


Figure 4.35: The effect of the primary network parameters on the mean reneging time of SUs vs N_2

sources, in 4.36 to be caused by the primary network characteristics. Fewer users balk as a result of SUs' opportunistic use of PCS.

$1/\lambda_1$ $C_x^2 > 1$ 4.28 4.16

In contrast to the secondary number of sources, Figure 4.37 illustrates how the principal subsystem rates affect the average residency duration of cognitive

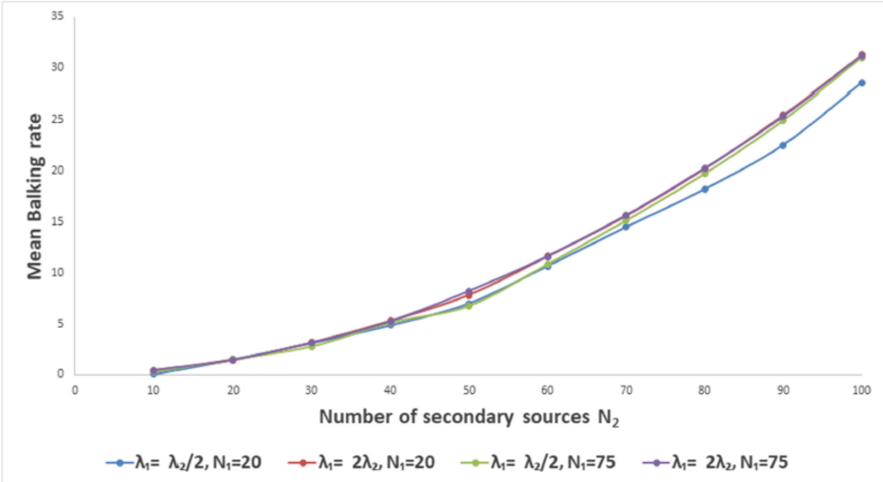


Figure 4.36: The effect of the primary network parameters on the mean balking rate of SUs vs N_2

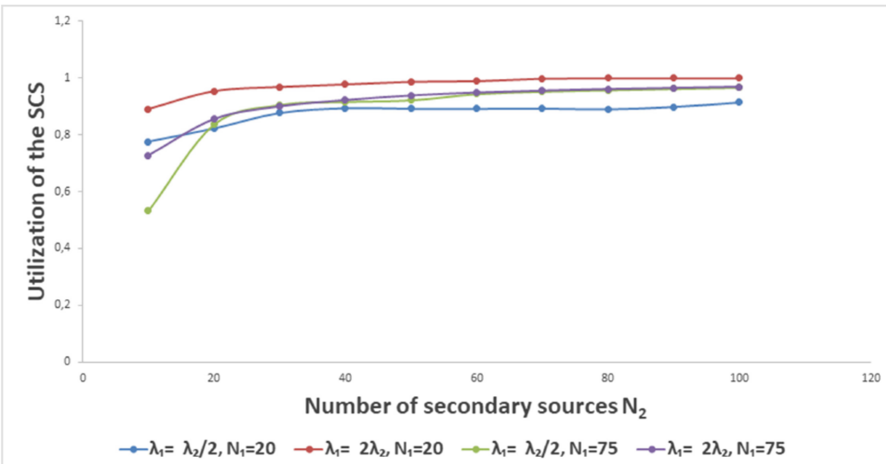


Figure 4.37: The effect of the primary network parameters on the utilization of SCS vs N_2

users. N_2 increases the secondary system's use, however, at a certain point it hits its limit and the server is completely utilized. When the primary arrival rate is at its greatest or lowest, $\lambda_1 = \lambda_2 * 2$ and $\lambda_1 = \lambda_2/2$respectively, a distinct difference may be seen.

In Figure 4.38, an impact could be seen when there are just a few sources $N_1 = 20$ and $N_2 = 10$. The mean response time of SUs vs N_2 may be affected

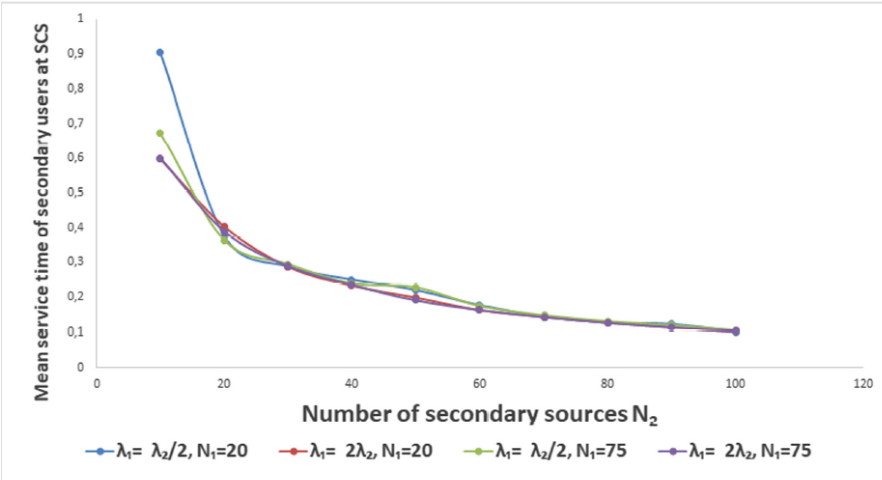


Figure 4.38: The effect of the primary network parameters on the mean service time of SUs vs N_2

by the main network characteristics, as seen in this graph.

5 Conclusion

The work of this dissertation is concluded and summarized in this chapter.

Using a simulation program and with the help of SimPack [19], we were able to model a retrial queuing system. In particular, discrete event simulation built on several algorithms was created to determine the efficacy of a cognitive radio network system with a finite number of sources.

In the realm of opportunistic spectrum access and the optimal utilization of wireless systems, cognitive radio networks stand out as the preferred choice.

Despite being a nascent field of study, research in this area is progressing rapidly. Chapter 2 aims to present cutting-edge cognitive radio networks, encompassing both primary and secondary sub-networks. The chapter is structured into three major parts, complementing the introduction and conclusion. The initial section defines the properties and functions of cognitive radio communications, laying the foundation crucial for building a network based on cognitive radio communication.

The subsequent part delves deeply into the design of a cognitive radio system, employing various illustrations. As expounded in this section, a Cognitive Radio Network (CRN) comprises multiple coexisting cognitive radio systems and other types of multi-radio communication systems. Conceptualized as a unique form of heterogeneous network integrating diverse communication technologies, CRNs not only detect nearby networks but also interface with communication infrastructure to enhance connectivity beyond conventional spectrum usage. Consequently, the third section of this chapter elucidates spectrum sensing approaches, utilizing figures and charts to categorize sensing techniques.

Chapter 3 opens with a summary of key theories on random processes and distributions. Highlighting distributions such as exponential, hypo-exponential, hyper-exponential, lognormal, Pareto, and Gamma, this section introduces key distribution functions used in our simulation program. The second part provides essential insights into simulation modeling fundamentals, preparing the groundwork for the discrete-event simulation elements discussed later. Section 3 of the literature introduces the related study, which incorporates the retry queueing system into cognitive radio communication.

The proposed approach in our research employs two bands to offer cognitive services to primary and secondary users. Prioritizing primary customers for services on licensed channels, an orbit is established on the secondary channel to assist secondary packets in case of busy servers. Chapter 4 aims to collect key variables and formulas for creating simulation software to model our system. Consequently, section 4 utilizes graphics to present flowcharts explaining program logic and illustrating the organization of the simulation.

We examine the effects of the inter-event time distributions on the key performance measures of the system with the use of the simulation tool built for this thesis. In certain instances, we also assess how effectively the main and secondary channel services are being used. The simulation software developed by Andrea Francini in [20] has a statistical class that can be used to estimate the mean and variance of the response time. The system's behaviour may be explained using a continuous-time Markov chain in a specific scenario where all times are exponentially distributed, and the key stationary performance characteristics can be determined. In contrast, we expand the model in our study by assuming that some of the inter-event times are not exponentially distributed. Therefore, The functionality of the batch mean methods, the primary routine of the statistical class that provides the estimate of the mean and variance, were provided in section 3.4.3. It involves segmenting the simulation run into several blocks of observations and, after a "warm-up" phase, providing an estimate of the mean for each block. This latter test is the first one run at the start of the simulation and is necessary to build the confidence interval for the mean and determine how close the estimate resembles the final expectation. It should be noted that for the output analysis of a steady-state simulation, the batch mean approach is the most often used confidence interval methodology.

Besides the basic exponential distribution, we were able to implement more distributions for the inter-arrival times, such as hypo-exponential, hyper-exponential, lognormal and more. The results of the key performance measures of the system are presented in this chapter. These research instances were mentioned:

- We dealt with a finite-source cognitive radio network where the low-priority subsystem is unreliable, where random breakdowns and repairs take place in both statuses busy and free. We aimed to examine how the key performance indicators of the system are affected by the servers' unreliability time distributions and failure appearance (busy or idle).

- We examined a finite-source cognitive radio network with unreliability and abandonment. We were able to assess the performance metrics of the system assuming that the abandonment times are generally distributed random variables with the same mean and different variance. The aim was to investigate the impact of the implemented distributions on the mean of the sojourn time. The results showed that the performance metrics are significantly impacted by the service time distributions, more precisely, the effect was caused by the squared coefficient of variation of these distributions.

- In section 4.4 we included the most known users' behaviours during waiting "balking" (refusing to enter the queue) and "reneging" (leaving the system after entering). These two phenomena were applied to the cognitive radio network. We assumed that the service time of SUs is generally distributed. The goal was to thoroughly examine how cognitivity and service time distributions affected the key performance measures of our system.

In light of cognitive radio technology, the purpose of this thesis is to link two different retrial queueing systems. This was motivated by earlier similar publications that evaluated the efficacy of systems of this kind using theoretical techniques. We developed a simulation program that, by allowing the non-exponential distributions and general situations, helped us to evaluate the efficiency of the system. Several sample cases were obtained by demonstrating the impact of the distribution of the inter-events times on the first and second moments of the response times with the use of SimPack and the simulation program.

6

Acknowledgements

I am indebted to so many, in particular I would like to start by expressing my gratitude to my supervisor, Prof. János Sztrik , for his invaluable advices, continuous support, and patience throughout my time as his student. His immense knowledge and plentiful experience have encouraged me all the time in my academic research. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

I would like to thank my colleague and close friend Dr. Hamza Nemouchi in addition to my supervisor, for helping me through my whole research. Without his crucial help, this work would have been much more difficult.

I wish in addition to thank the Stipendium Hungaricum for their financial assistance. They have been extremely beneficial for me to finish my education. Immense gratitude as always to my family for their love and support.

Mohamed Hedi Zaghouani

Debrecen, Hungary, 2023

Bibliography

[1] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty. Adaptnet: an adaptive protocol suite for the next-generation wireless internet. *IEEE Communications Magazine*, 42(3):128 – 138, 2004.

[2] B. Almási, T. Bérczes, A. Kuki, J. Sztrik, and J. Wang. Performance modeling of finite-source cognitive radio networks. *Acta Cybernetica*, 22(3):617–631, 2016. `doi:10.14232/actacyb.22.3.2016.5`.

[3] J.R. Artalejo. New results in retrial queueing systems with breakdown of the servers. *Statistica Neerlandica*, 48:23–36, 1994.

[4] J.R. Artalejo. Retrial queues with a finite number of sources. *J. Korean Math. Soc*, 35:503–525, 1998.

[5] J.R. Artalejo and A. Gomez-Corral. Information theoretic analysis for queueing systems with quasi-random input. *Mathematical and Computer Modelling*, 22:65–76, 1995.

[6] A. Azarfar, J. F. Frigo, and B. Sansó. Analysis of cognitive radio networks based on a queueing model with server interruptions. *in Proc. IEEE Int. Conf. Commun. (ICC)*, pages 1703–1708, 2012.

[7] A. Azarfar, J. F. Frigo, and B. Sansó. Dynamic selection of priority queueing discipline in cognitive radio networks. *in Proc. IEEE Veh. Technol. Conf. (VTC Fall), Québec, QC, Canada*, pages 1–5, 2012.

[8] V. Bassoo and N. Khedun. Improving the quality of service for users in cognitive radio network using priority queueing analysis. *IET Commun.*, 10(9):1063–1070, 2016.

[9] M. M. Buddhikot, P. Kolody, S. Miller, K. Ryan, and J. Evans. Dimsumnet: new directions in wireless networking using coordinated dynamic spectrum access. *IEEE WoWMoM*, pages 78 – 85, 2005.

[10] E. Carlstein and D. Goldsman. The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *Annals of Statistics*, 14:1171–1179, 1986.

[11] W. S. Chang and W. M. Jang. Spectrum occupancy of cognitive radio networks: A queueing analysis using retrial queue. *IET Netw.*, 3(3):218–227, 2014.

[12] E. Jack Chen and W. David Kelton. A procedure for geneating batch means confidence intervals for simulation: Checking independence and normality. *SIMULATION*, 83(10):683–694, 2007.

[13] K. C. Chen and R. Prasad. *Cognitive Radio Networks*. Wiley, 2009.

[14] Natasha Devroye, Mai Vu, and Vahid Tarokh. Cognitive radio networks. *IEEE Signal Processing Magazine*, 25(6):12–23, 2008.

[15] V.L. Dragieva. Single-line queue with finite source and repeated calls. *Problems of Information Transmission*, 30:283–289, 1994.

[16] A. N. Dudin, M. H. Lee, O. Dudina, and S. K. Lee. Analysis of priority retrial queue with many types of customers and servers reservation as a model of cognitive radio system. *IEEE Trans. Commun.*, 65(1):168–199, 2017.

[17] Spectrum Policy Task Force FCC. Report of the spectrum efficiency working group. Washington, DC, 11 2002.

[18] G. S. Fishman. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer, 2001. `doi:10.1007/978-1-4757-3552-9`.

[19] Paul A. Fishwick. Getting started with simulation programming in c and c++. *Winter Simulation Conference*, pages 154–162, 1992.

[20] A. Francini and F. Neri. A comparison og methodologies for the stationary analysis of data gathered in the simulation of telecommunication networks. pages 166–122, February 1996.

[21] G. Ganesan and Y. G. Li. Cooperative spectrum sensing in cognitive radio networks. *IEEE DySPAN*, pages 137 –143, 2005.

[22] N. Gharbi and C. Dutheillet. An algorithmic approach for analysis of finite-source retrial systems with unreliable servers. *Computers and Mathematics with Applications*, 62(6):2535–2546, 2011.

[23] N. Gharbi, L. Mokdad, and J. Ben-Othman. A performance study of next generation cellular networks with base stations channels vacations. *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2015.

[24] N. Gharbi, Besma Nemmouchi, L. Mokdad, and J. Ben-Othman. The impact of breakdowns disciplines and repeated attempts on performances of small cell networks. *Journal of Computational Science*, 5(4):633–644, 2016.

[25] A. Ghasemi and E. S. Sousa. Agility improvement through cooperative diversity in cognitive radio networks. *GLOBECOM*, pages 2505–2509, 2005.

[26] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley, 4 edition, 2008.

[27] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal On Selected Areas In Communications*, 23(2):201 – 220, 2005.

[28] J. Heo, J. Shin, J. Nam, Y. Lee, J. G. Park, and H. S. Cho. Mathematical analysis of secondary user traffic in cognitive radio system. *in Proc. IEEE 68th Veh. Technol. Conf., Calgary, Canada*, pages 1–5, 2008.

[29] D.J. Houck and W.S. Lai. modelling and analysis of hybrid fibercoax systems. *Computer Networks and ISDN Systems,*, 30:821–834, 1998.

[30] Li Hui and Yang Tao. A single server retrial queue with server vacations and a finite number of input sources. *European Journal of Operational Research*, 85:149–160, 1995.

[31] L. Ikhlef, O. Lekadir, and D. Aissani. Mrspn analysis of semi-markovian finite source retrial queues. *Annals of Operations Research*, 247(1):141–167, 2016.

[32] O. Ileri, D. Samardzija, and N. B. Mandayam. Demand responsive pricing and competetive spectrum allocation via spectrum server. *IEEE DySPAN*, pages 194 – 202, 2005.

[33] W. M. Jang and W. S. Chang. Queueing with transmission rate selection for cognitive radio networks in nakagami-m fading. *in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), San Diego, CA, USA*, pages 247–251, 2013.

[34] G.K. Janssens. The quasi-random input queueing system with repeated attempts as a model for collision-avoidance star local area network. *IEEE Transactions on Communications*, 45:360–364, 1997.

[35] KAM Kotb and Hassnaa A El-Ashkar. Quality control for feedback m/m/1/n queue with balking and retention of reneged customers. *Filomat*, 34(1):167–174, 2020.

[36] Attila Kuki, Tamas Berczes, Janos Sztrik, and Adam Toth. Reliability analysis of a two-way communication system with searching for customers. In *2019 International Conference on Information and Digital Technologies (IDT)*, pages 260–265. IEEE, 2019.

[37] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis.* McGraw-Hill College, second edition, 1991.

[38] L. Liu, X. Jin, G. Min, J. Hu, and K. Li. Modelling and analysis of dynamic spectrum access in cognitive radio networks with self-similar traffic. *in Proc. IEEE Global Telecommun. Conf., Miami, FL, USA*, pages 1–5, 2010.

[39] Myron H MacDougall. *Simulating computer systems: techniques and tools.* MIT press, 1987.

[40] M.A. Mehmet, J.F. Hayes, and A.K. Elhakeem. Traffic analysis of a local area network with star topology. *IEEE Transactions on Communications*, 36:703–712, 1988.

[41] J. Mitola and G. Maguire. Cognitive radio: Making software radios more personal. *IEEE Personal Communications*, 37(10):13–18, 1999. `doi:10.1109/98.788210`.

[42] Anatoly Nazarov, János Sztrik, and Anna Kvach. A survey of recent results in finite-source retrial queues with collisions. In *Information Technologies and Mathematical Modelling. Queueing Theory and Applications*, pages 1–15. Springer, 2018.

[43] Hamza Nemouchi and János Sztrik. Performance evaluation of finite-source cognitive radio networks with non-reliable services using simulation. In *Annales Mathematicae et Informaticae*, volume 49, pages 109–122. Eszterházy Károly University Institute of Mathematics and Informatics, 2018.

[44] H. Ohmura and Y. Takahashi. An analysis of repeated call model with a finite number of sources. *Electronics and Communications in Japan*, 68:112–121, 1985.

[45] B. Oklander and M. Sidi. Modeling and analysis of system dynamics and state estimation in cognitive radio networks. *in Proc. IEEE 21st Int. Symp. Pers., Indoor Mobile Radio Commun.Workshops, Istanbul, Turkey*, pages 49–53, 2010.

[46] A. Sahai, N. Hoven, and R. Tandra. Some fundamental limits in cognitive radio. *Allerton Conference on Communication, Control, and Computing*, pages 1662 –1671, 2004.

[47] H. Schwetman. Using csim to model complex systems. pages 246–253, 01 1989. `doi:10.1109/WSC.1988.716152`.

[48] I. Suliman and J. Lehtomäki. Queueing analysis of opportunistic access in cognitive radios. *in Proc. 2nd Int. Workshop Cogn. Radio Adv. Spectr. Manage., Aalborg, Denmark*, pages 153–157, 2009.

[49] Ádám Tóth and János Sztrik. Simulation of finite-source retrial queuing systems with collisions, non-reliable server and impatient customers in the orbit. In *ICAI*, pages 408–419, 2020.

[50] P. Tran-Gia and M. Mandjes. Modeling of customer retrial phenomenon in cellular mobile networks. *IEEE Journal of Selected Areas in Communications*, 15:1406–1414, 1997.

[51] J. Tyser. *Object-Oriented Computer Simulation of Discrete-Event Systems*. Springer Science, LLC, first edition, 1999.

[52] T. Weiss and F. Jondral. Spectrum pooling: An innovative strategy for the enhancement of spectrum efficiency. *IEEE Communication Magazine*, 42:S8 –S14, 2004.

[53] A. J. White, J. W. Schmidt, and G. K. Bennett. *Analysis of Queueing Systems.* Academic Press, INC, first edition, 1975.

[54] Y. Xiao and F. Hu. *Cognitive Radio Networks.* CRC Press, 2008.

[55] S. A. Zekavat and X. Li. User-central wireless system: ultimate dynamic spectrum allocation networks. *IEEE DySPAN*, pages 82 – 87, 2005.

[56] M. Zhang, S. Jiang, G. Wei, and H. Wang. Performance analysis of the cognitive radio network with a call level queue for secondary users. *in Proc. 5th Int. Conf. Wireless Commun., Netw. Mobile Comput., Beijing, China*, pages 1–4, 2009.

[57] Y. Zhang, J. Zhend, and H. Chen. *Cognitive Radio Networks: Architectures, Protocols, and Standards.* CRC Press, 2016.

[58] Djamila Zirem, Mohamed Boualem, Karima Adel-Aissanou, and Djamil Aïssani. Analysis of a single server batch arrival unreliable queue with balking and general retrial time. *Quality Technology & Quantitative Management*, 16(6):672–695, 2019.

My publications

List of Papers [J]:

J1 Zaghouani, M.H., Sztrik, J., and Uka A.,
Simulation of the performance of Cognitive Radio Networks with unreliable servers,
Ann. Math. Inform. 52, 255-265, 2020. ISSN: 1787-5021.

J2 Nemouchi, H., Zaghouani, M.H., and Sztrik, J.,
Analysis of Cognitive Radio Networks with Balking and Reneging,
Distributed Computer and Communication Networks: Control, Computation, Communications / Vishnevskiy Vladimir M, Springer International Publishing Ag, Cham, 3-13, 2021. ISBN: 9783030925062

J3 Zaghouani, M. H., Nemouchi, H., Sztrik, J.,
Reliability analysis of Cognitive Radio Networks with balking and reneging,
The International Conference on Digital Technologies (IDT 2021), IEEE, [s.l.], 212-215, 2021. ISBN: 9781665436922.

J4 Nemouchi, H., Zaghouani, M.H., and Sztrik, J.,
Simulation Analysis in Cognitive Radio Networks with Unreliability and Abandonment,
Information Technologies and Mathematical Modelling. Queueing Theory and Applications / Alexander Dudin; Anatoly Nazarov; Alexander Moiseev, Springer International Publishing Ag, Cham, 31-45, 2021, (Communications in Computer and Information Science, ISSN 1865- 0929 ; 1391.) ISBN: 9783030722463.

J5 Zaghouani, M. H., Sztrik, J., Uka, A.,
Reliability Analysis of Cognitive Radio Networks,
Information technologies and mathematical modeling (ITMM-2019) : Proceedings of the XVIII International Conference named after A.F. Terpugov 26-30 June 2019. Eds.: A. A. Nazarov, S. P. Moiseeva, A. Matrosova, E. Lisovskaya, NTL, Tomsk, 110-114, 2019. ISBN: 9785895036297.

Software developed during PhD:

- *Simulation programs:* Based on SimPack with C/C++ language

My publications

List of Conference proceedings [C]:

C1 Zaghouani, M. H., Nemouchi, H., Sztrik, J.,
Performance investigation of reverse balking in cognitive radio networks using simulation,
Informational technologies and mathematical modelling (ITMM-2021): Proceedings of the 20th International Conference named after A. F. Terpugov (2021 December, 1-5) / Nazarov, A. A., Moiseeva, S. P., Moiseev, A. N, Tomsk State University Publishing, Tomsk, 145-149, 2022. ISBN: 9785907572201.

C2 Zaghouani, M. H., Nemouchi, H., Sztrik, J.,
Performance modeling of finite-source cognitive radio networks with reverse balking and reneging using simulation,
2022 IEEE 2nd Conference on Information Technology and Data Science (CITDS). Proceedings. Ed.: Fazekas István, IEEE Computer Society, Washington, 330-333, 2022. ISBN: 9781665496544.

C3 Nemouchi, H., Zaghouani, M. H., Sztrik, J.,
Simulation analysis in cognitive radio networks with unreliability and abandonment,
Information technologies and mathematical modelling (ITMM-2020) : Proceedings of the 19th International Conference named after A. F. Terpugov (2020 December, 2-5). Eds.: A. A. Nazarov, S. P. Moiseeva, A. N. Moiseev, M. P. Farhadov, E. Yu. Lisovskaya, Scientific Technology Publishing House, Tomsk, 110-114, 2021. ISBN: 9785895036471.

C4 Nemouchi, H., Zaghouani M.H., and Sztrik,
The Impact of Servers Reliability on the Characteristics of Cognitive Radio Systems,
Proceedings of 1st Conference on Information Technology and Data Science (CITDS 2020) / editors István Fazekas, András Hajdu, Tibor Tómács, CEUR, Debrecen, 151-167, 2021, (CEUR Workshop Proceedings, ISSN 1613-0073 ; 2874).

C5 Zaghouani, M. H., Sztrik, J.,
Performance evaluation of finite-source Cognitive Radio Networks with impatient customers,

Ann. Math. Inform. 51, 89-99, 2020. ISSN: 1787-5021. DOI: http://dx.doi.org/10.33039/ami.2020.07.004.

C6 Zaghouani, M. H., Sztrik, J., Melikov, A.,
Reliability Analysis of Cognitive Radio Networks,
Proceedings of the International Conference on Information and Digital Technologies 2019, IEEE, [s.l.], 574-579, 2019.