MULTI-DESTINATION RENDEZVOUS IN COGNITIVE RADIO NETWORKS

by

Tamer Samak

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2017

Approved by:

____
Dr. Jiang (Linda) Xie

____
Dr. Thomas Weldon

____
Dr. Tao Han

ABSTRACT

TAMER SAMAK. Multi-destination rendezvous in cognitive radio networks.
(Under the direction of DR. JIANG (LINDA) XIE)

The radio spectrum is a limited natural resource and the increase of wireless devices operating in the unlicensed bands of the spectrum has led to its overcrowding resulting in poor performance of those radio devices. Moreover, the static spectrum allocation has resulted in low spectrum efficiency in the licensed bands. Those factors, in addition to the recent growth in data greedy apps and their need for more bandwidth, have made enhancing the usage of radio spectrum a highly desirable objective. Cognitive radio (CR) networks are designed based on the concept of dynamic spectrum sharing where CR users can opportunistically share the radio resources that might have equal or unequal access rights. Intelligent CR device can sense and identify vacant areas or spectrum holes that can be used for communications thus maximizing the utilization.

In CR networks, rendezvous is when two secondary users tune to the same frequency channel simultaneously so that they can communicate with each other. The rendezvous delay, a.k.a the time to rendezvous (TTR), has been a highly focused topic for research. Most existing papers tried to reduce the TTR between a pair of secondary users (SUs), a source and a destination. To the best of our knowledge, no paper has previously considered the scenario of a SU sender having different packets in its buffer for multiple destinations. Those who approached a similar scenario relied on a common control channel or the presence of multiple radios. In this research, we consider blind rendezvous using a single radio. We propose a new rendezvous protocol to handle the multiple destination scenario to decrease the overall TTR and increase the throughput, thus enhancing the overall performance of the CR network. Extensive simulations are carried out to demonstrate the proposed protocol performance.

ACKNOWLEDGEMENTS

This work would not have been possible without the help, support and patience of my advisor, Dr. *Jiang (Linda) Xie*, not to mention her advice, vital encouragement and support.

I am most grateful to the *Fulbright Foreign Student Program* , the Fulbright scholarship provided me with the finances which have been a valuable and reliable resource to complete this work.

DEDICATION

Above all, I would like to thank *GOD* who made all things possible. Without His blessings and continuous support, I could not have completed this work.

I would like to acknowledge and extend my heartfelt gratitude to the following persons who have made the completion of this work possible.

I want to thank my dear *Father*, may the Lord repose his soul in heavens, my dear *Mother* and my dearest brother *Elhamy* for their great patience and love at all times. They have always been there for me and I have been blessed by their personal support throughout my life for which my mere expression of gratitude does not suffice.

Last, but by no means least, I would like to thank my friends and my colleagues for their kindness, friendship, encouragement and support.

Thank you all.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Background

Nowadays spectrum-based technologies are of growing interest among research within academia, industry and spectrum policy makers, and are increasingly the real drivers of innovation in telecommunications. The radio spectrum is a limited natural resource, and the increase and success of wireless devices operating in unlicensed bands of the spectrum has led to overcrowding of those bands leading to poor performance of these radio devices. Moreover, the static spectrum allocation has resulted in low spectrum efficiency in licensed bands.

Consequently with the recent growth in data apps and the need of more bandwidth to accommodate such data greedy apps, enhancing the usage of the radio spectrum became a highly desirable objective

"Cognitive Radio" has emerged as a new design for next generation wireless networks. Cognitive radio networks are designed based on the concept of dynamic spectrum sharing where cognitive radio users can opportunistically share the radio resources that might have equal or unequal access rights. In cognitive radio networks secondary (unlicensed) users are allowed to transmit as long as they do not degrade primary (licensed) users' communication.

A conventional, hardware-based wireless device can access only one area of the radio spectrum, but an intelligent cognitive radio device can sense and identify "white spaces", "spectrum holes" as shown in Figure 1, or vacant areas, in the spectrum that can be used for communications thus maximizing the utilization of the limited radio bandwidth.

In such environment where secondary users have no previous knowledge of the channels that will be used for their communication, it is hard for the sender and the receiver to find each other on the same channel which is known as the rendezvous. Some earlier studies suggested the use of a common control channel where users can communicate control information such as the channel on which the communication will occur between secondary users but such suggestion proved to suffer many problems, which led to researchers trying to avoid the use of a common control channel to coordinate the rendezvous of secondary users, such is known as *blind rendezvous*.

1.2 Problem Statement and Research Motivation

For blind rendezvous between secondary users, channel hopping is a very common technique to achieve the rendezvous in such cognitive environment between a sender and a receiver. Many studies have researched the hopping sequence and many algorithms were proposed to ensure that the rendezvous between two secondary users would occur if they have at least one channel in common available for their communication. The rendezvous delay, also known as the *time to rendezvous (TTR)*, has been a highly focused topic for research. Most papers tried to reduce the TTR between a pair of secondary users (SUs) using various techniques, among which tampering with the hopping sequence is a significant one.

Most of the papers have considered the scenario of a pairwise rendezvous where a sender has information to transmit to a single destination. However, none of the existing papers have considered the multi-destination rendezvous scenario. Some papers approached close scenarios relying on the presence of multiple radios or trying to unify the channel hopping sequence for all users. Using multiple radios would increase the cost. On the other hand, trying to unify the channel hopping sequence for all users results in the overhead of spreading information between the SUs in the process of building a unified hopping sequence. Moreover the available channel sets of different SUs dynamically change all the time so the unified built sequence will need to be revisited revisited all the time in such cognitive environment which would result in more overhead.

To the best of our knowledge, no papers have previously considered the scenario of a SU sender having different packets in its buffer for multiple destinations. That means the multi-destination scenario would be always treated in a traditional first-in-first-out way where destinations would be handled sequentially resulting in severe issues. Handling destinations in such sequential way with the usual rendezvous delay would result in extremely longer mean time to rendezvous (MTTR). Longer average rendezvous delays would result in lower throughput as longer time would be wasted for the sake of achieving rendezvous instead of actually transmitting data. Obviously, lower throughput in the network means performance degradation and lower quality of service (QoS). In a cognitive radio network where channels are not always available for SUs and where power restrictions exist on SUs' transmissions so as not to degrade PUs performance, the QoS is already lower than that of a similar primary users network. Therefore, QoS in a cognitive radio network should not be allowed to further suffocate.

1.3 Contributions

We propose a new rendezvous protocol to handle the multi-destinations scenario, without having to use more than a single radio interface. Using this protocol, the contents of the transmission buffer will be analyzed. In addition, the transmission of different messages to a group of destinations will be handled as one whole, instead of being handled sequentially as the traditional first-in-first-out technique.

Simulations of the performance of the new protocol demonstrate a decrease in the overall time to rendezvous (TTR) and increase in the throughput, thus an overall enhancement in the performance of the cognitive radio (CR) network.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows. In Chapter 2, we shed more light on the background of cognitive radio networks and related work. In Chapter 3, Our proposal is explained showing the scheme design details and considerations. In Chapter 4, we show the performance evaluation of our proposed scheme. In chapter 5, we provide a summary of the contributions and present a proposal for future work.

CHAPTER 2: BACKGROUND AND RELATED WORK

2.1 Motivation and Background of Cognitive Radio Networks

2.1.1 Motivation of Cognitive Radio

In the age of information, data communication has become a vital part of our daily lives and also an integral part of all aspects of modern society. Today data communication is essential for financial transactions, social interactions, education, ..etc. As a result, the number of data applications and the number of their users are significantly growing. In addition, many diverse services have evolved such as voice over IP telephony (VoIP), web browsing, instant messaging, multimedia messaging, video streaming, ..etc., each with different performance requirements in regards to bandwidth/data rate, latency, power consumption, quality of service, ..etc. Moreover, there is a rapid growth in the number of new users and the number of new wireless services being offered as shown in the below Figure 2.1.

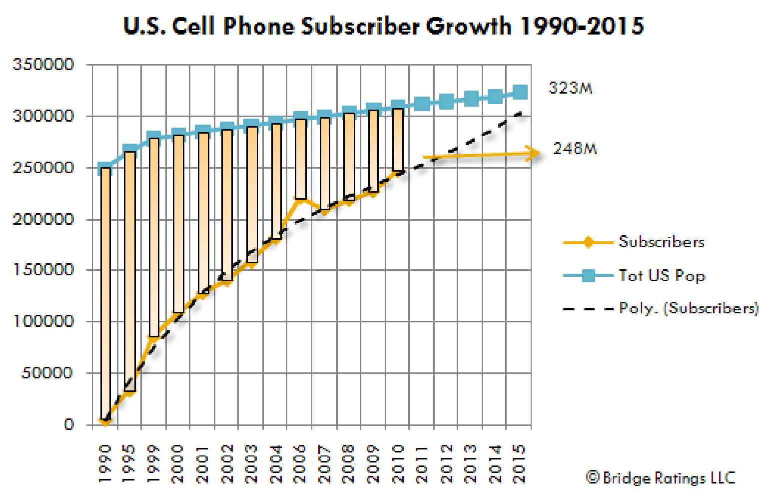


Figure 2.1: US Cell Phone Subscriber Growth 2015 [1]

In order to satisfy such requirements and to accommodate the rapid growth in the wireless devices, more bandwidth is required. As mentioned earlier, The radio spectrum is a limited natural resource, and the increase and the success of wireless devices operating in the unlicensed bands of the spectrum has led to their overcrowding leading to poor performance of these radio devices. Moreover, the static spectrum allocation - Figure 2.2 - has resulted in low spectrum efficiency in licensed bands. Recent measurements by FCC show that up to 85% of the allocated spectrum is not efficiently utilized and with the recent growth in data apps and the need of more bandwidth to accommodate such data greedy apps, enhancing the usage of the radio spectrum has become a necessity.

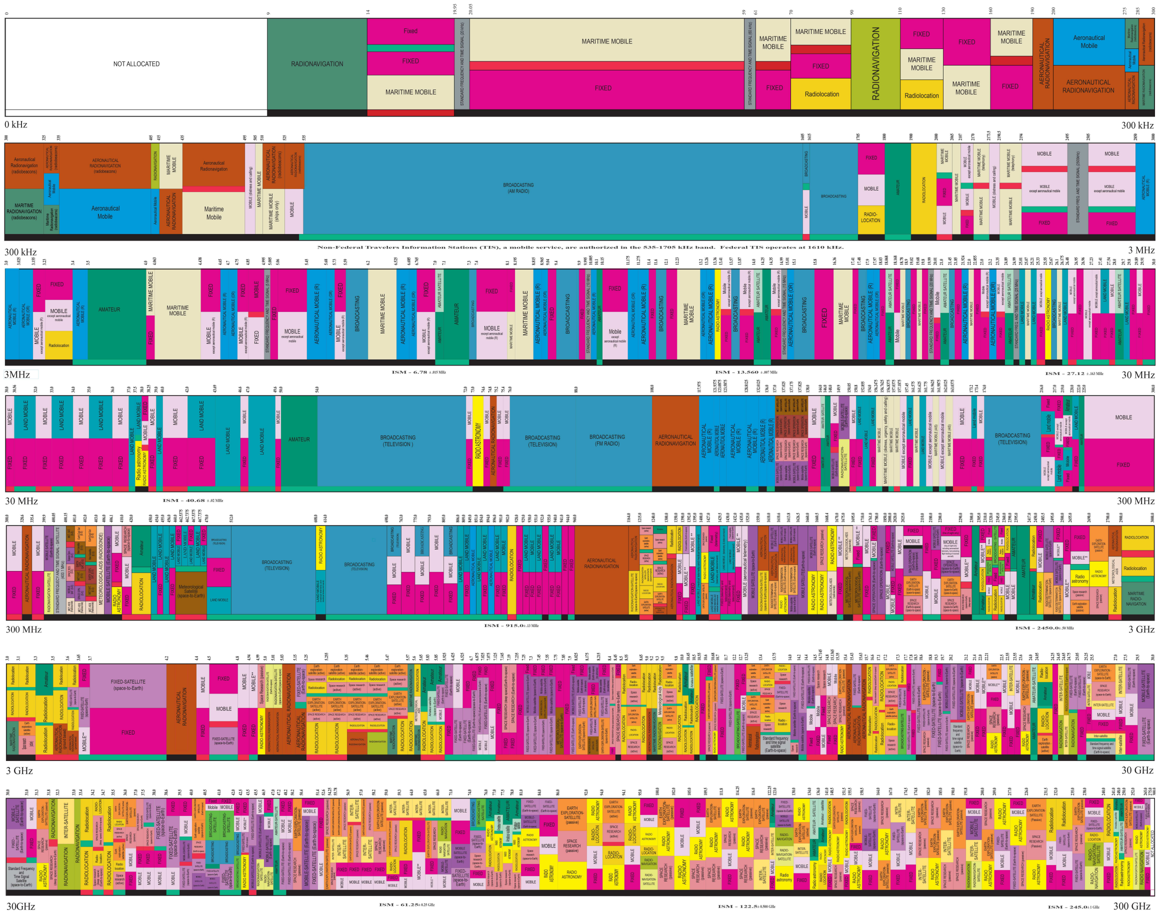


Figure 2.2: United States Frequency Allocations as of January 2016; The Radio Spectrum [2]

Conventional wireless communication devices are constrained in operation. They serve only fixed applications (e.g. LTE, WLAN, Bluetooth) and they have fixed modes of operation (e.g. modulation scheme, data rates, power levels). Moreover they can access only fixed frequency bands of operation. With that mentioned, it is obvious that conventional wireless communication devices with their constrained capabilities cannot contribute to solving the spectrum underutilization problem which implies that a different tier of wireless communication devices is needed. [9]

2.1.2 Cognitive Radio Networks

Cognitive radio has emerged as a new design for next generation wireless networks. A cognitive radio is intelligent wireless communications system based on SDR technology. The Software defined radio (SDR) is an enabling technology that allows the Cognitive radio device to operate with much less constraints than those of the conventional wireless device.

In the strict definition, “A cognitive radio (CR) is a radio that can change its transmitter parameters based on interaction with the environment in which it operates.” – FCC

We define primary users (PU) as licensed users, those users or entities who have payed for exclusive access of some parts of the spectrum on full time basis whether they are actually using that part of the spectrum or not - *spectrum owners*. Secondary users (SU) are the *unlicensed users/entities* who have the capabilities to access wider parts of the spectrum whether licensed or unlicensed, and those users/entities have lower priority accessing the licensed parts of the spectrum.

Cognitive radio networks are designed based on the concept of dynamic spectrum sharing where cognitive radio users can opportunistically share the radio resources that might have equal or unequal access rights. In cognitive radio networks, secondary (unlicensed) users are allowed to transmit as long as they do not degrade primary (licensed) users’ communications

An intelligent cognitive radio device can sense and identify “white spaces”, “spectrum holes" as shown in Figure. (again), or vacant areas, in the spectrum that can be used for communications thus maximizing the utilization of the limited radio bandwidth. [9]

2.1.3 Cognitive Radio Network Infrastructure

From a control point of view, the cognitive radio networks can be classified into either of two models:

a. Centralized: where a central entity, most commonly a base station, Figure 2.3, controls the sharing and allocation of spectrum and communication resources. In such model, network nodes are synchronized and there are common coordinated Quiet Periods (QP) where all CR-nodes halt their transmission and listen (sense) the spectrum to detect Primary Users' activity. Gathered information about PU activity is shared among nodes through a database which holds information about PU activity on all scanned channels, thus each node can rely on that database for information about spectrum holes rather than relying on self-results obtained through sensing spectrum bands. A common example for such network model is the one using the IEEE 802.22 MAC standard.

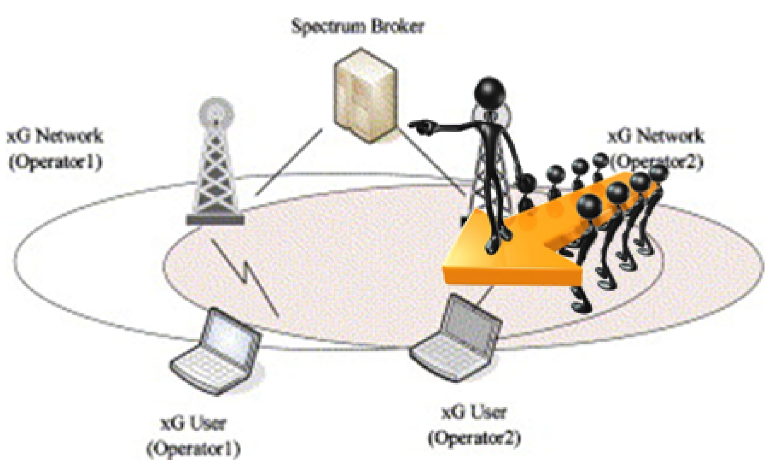


Figure 2.3: Centralized Infrastructure [3]

b. Distributed: where there is no central entity responsible for control. Nodes operate in an ad-Hoc fashion, as shown in Figure 2.4, where every nodes contends to access the spectrum, most commonly using the CSMA/CA access technique. Such model supports that the node can operate both in an asynchronous mode or in synchronous mode. For synchronous mode, synchronization can take place through broadcasting a beacon in a common control channel. Quiet Periods in such model are either coordinated or uncoordinated throughout the network, thus each node can choose when to transmit and when to become silent to listen for PU activity. Of course for such model, information about spectrum holes is collected by each node through sensing of the spectrum bands

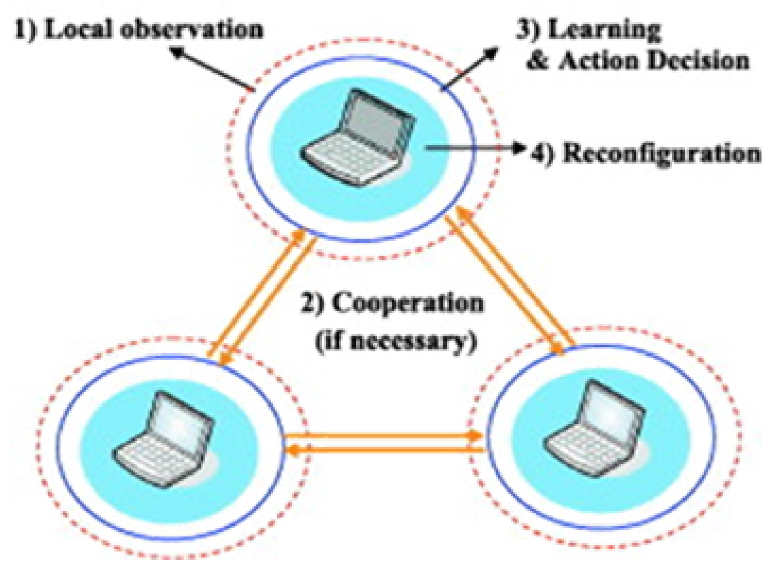


Figure 2.4: Distributed Infrastructure [3]

2.1.4 Cognitive Radio Medium Access Control

Cognitive radio Networks, as any other conventional network, can be modeled using the 7-Layers OSI model or the TCP/IP model but taking into consideration that Layer 2 (MAC layer) and Layer 1 (the physical layer) both operate in close collaboration and are tightly bonded, as shown in Figure 2.5, for a CRN successful operation. CR-MACs must support cross layer design for secondary system performance enhancements.

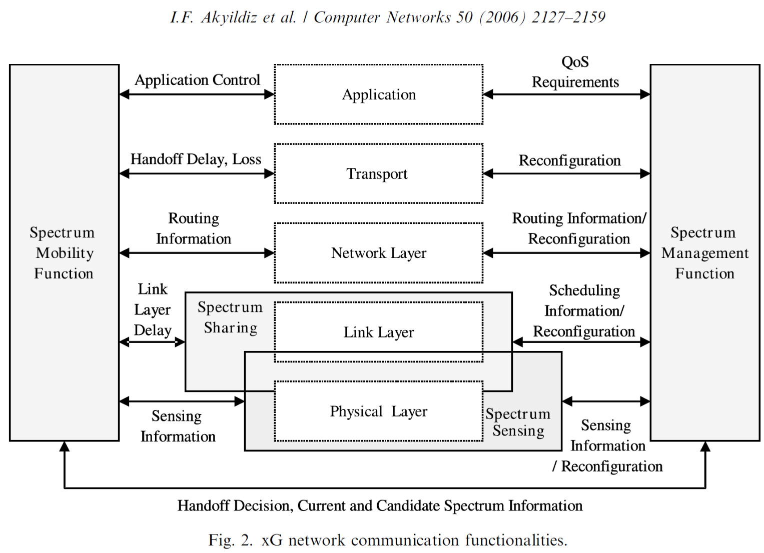


Figure 2.5: Close Collaboration in lower layers of CR network [4]

Conventional MAC protocols were designed mainly for efficient utilization of the available communication resources over statically allocated spectrum which is constantly available and were classified mainly based on the access scheme. CR-MAC protocols are designed for efficient utilization of the available spectrum for secondary usage, protection of the primary users by avoiding harmful interference to them and avoiding colliding with them as well, also fast recovery of the spectrum variability. CR-MACs have to deal with variable spectrum availability varying in time/space/frequency. Among the generic CR-MAC functionalities are spectrum sensing, spectrum sharing and control channel management with their aspects, as shown in Figure 2.6, which will be discussed in the following sections

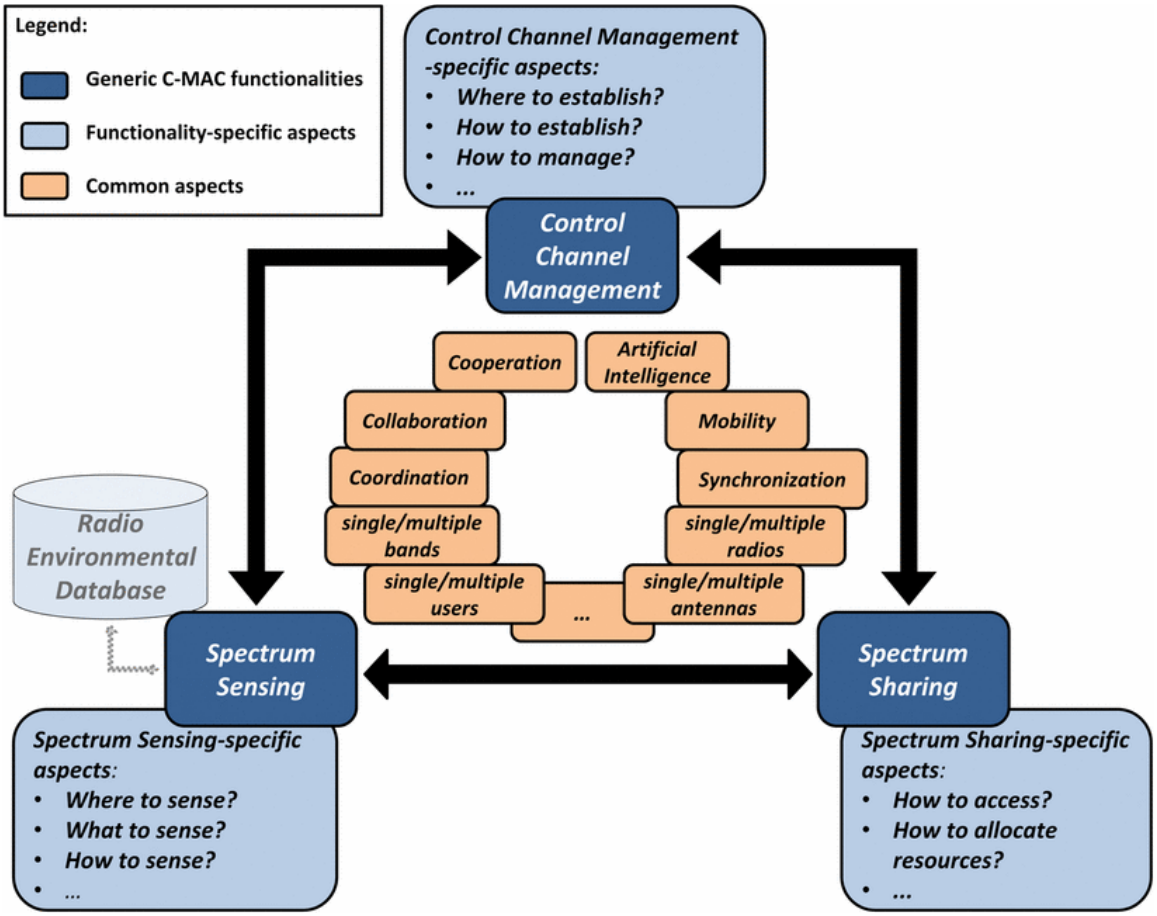


Figure 2.6: Close Collaboration in lower layers of CR network [5]

2.1.4.1 Spectrum Sensing

A. When to Sense

- Reactive Sensing: Sensing for new PU-free channels occurs reactively when the PU appears on the current SU channel, or when the SU moves to a new location and therefore has to look for a new channel, or when needed to increase the reliability on a certain PU channel
- Proactive Sensing: occurs to have an estimate and prediction of PU activity patterns, also when a statistical model of channel availability / occupancy is desired

IEEE 802.22 MAC standard which was deployed in the CRN operating on the

U.S. TV bands [54 - 865 MHz], has specified fast sensing as its proactive sensing mode, also specified fine sensing as its reactive sensing mode, as shown in Figure 2.7, where sensing takes place for extended intervals to allow for the application of advanced signal detection techniques to detect the PU, e.g. feature detection.

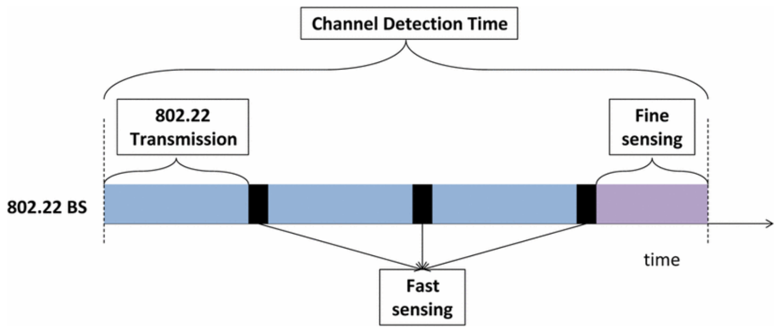


Figure 2.7: Fine versus fast sensing [5]

B. How to detect

- Blind: blind detection approach requires no prior knowledge about type/ structure of PU network and signals. This approach employs either the energy detection or the autocorrelation and Higher-Order-Statistical detection (HOS) detection. The energy detection is where the sensed power over the spectrum band is compared to a threshold to determine whether PU is present. Although this is an easy process to detect a PU, it results in poor performance as false alarms about PU presence can result from any other form of energy present even if it was a noise source. The autocorrelation and HOS provides better PU detection results. HOS was incorporated in the IEEE 802.22 standard

- Feature Detection: which relies on matched filter and cyclostationary detection of the cyclic or periodic form (similar to sine and cosine and their harmonics) of the PU modulated signal. Feature detection can also distinguish PU from SU signals on the cost of complexity. Feature detection requires more time for detection and needs higher computational power to implement

- Cooperative environments: where nodes cooperate to detect PU by sharing their sensing output, then each node can get a more assured result of the presence of PU.

  Though such cooperation results in better detection, specially with hidden terminal issue, it incurs higher control bandwidth cost

  Radio Environment Maps (REM) is a recent approach that was suggested in cooperative environments where sensing outputs from nodes will be used to build a map showing the history of PU activity in a certain region and predicting PU activity patterns. Such maps will be stored in regional databases which will incur additional costs

C. Where to Sense

- Single vs. Multiple Radio: a wireless device with more than one transceiver can scan more than one PU band at the same time which is more efficient locating spectrum opportunities. Of course this comes at the cost of extra hardware leading to increase the price of the wireless device.

- In-band vs. Out-of-band: In-band sensing occurs on the same channel where secondary data transfer concurrently takes place thus avoiding collisions with PUs and SUs as well. Out-of-band sensing can take two forms, split-phase which occurs in cyclic fashion with data transmission, as shown in Figure 2.8 and concurrent which can only occur if the wireless device possesses multi-antenna/radio capability. Concurrent out-of-band sensing achieves better efficiency but at the hardware extra cost.

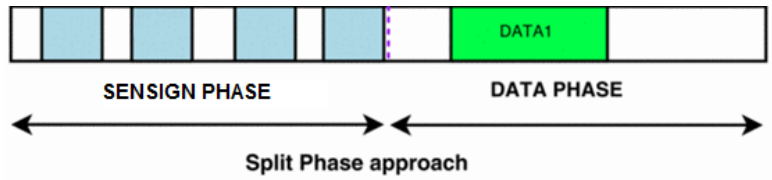


Figure 2.8: Split phase sensing [6]

D. How Long to Sense

A SU can stop sensing at the detection of an idle PU channel, N idle PU channels or when the optimal sensing stopping rule (economy science) is realized that is when the current reward is more than the expected reward.

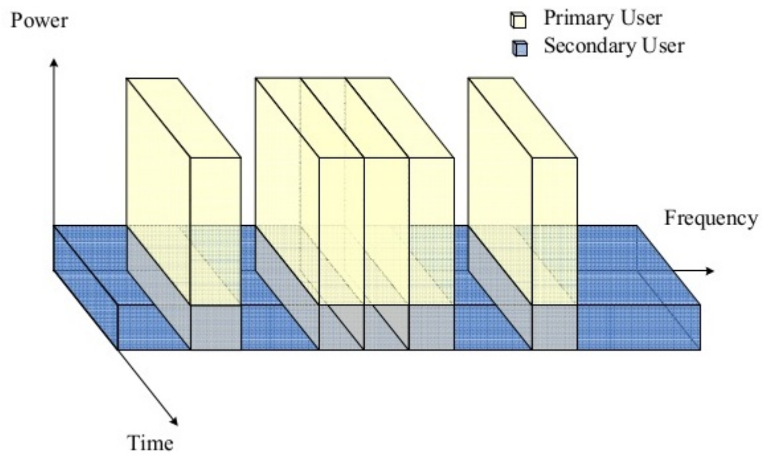
2.1.4.2 Spectrum Sharing

A. How to Access

Many access techniques can be utilized each multiplexing users in a different dimension, CSMA, OFDMA using different frequencies, SDMA differ in space, CDMA using different codes, TDMA/FDMA and DFH both using different time/frequency combinations. Among these CSMA is the only natural "cognitive" (opportunistic) multiple access scheme. It avoids collisions between the involved radios (SU to PU or among SUs) by adapting the contention windows, back-off durations.

B. How to Share (Sharing Modes)

i. Underlay: SUs must not to exceed the Max. Noise threshold of PUs so as not to cause harmful interference or degrade their communication. To achieve that, SUs communication must be spread using sophisticated spread spectrum techniques as shown in Figure 2.9.

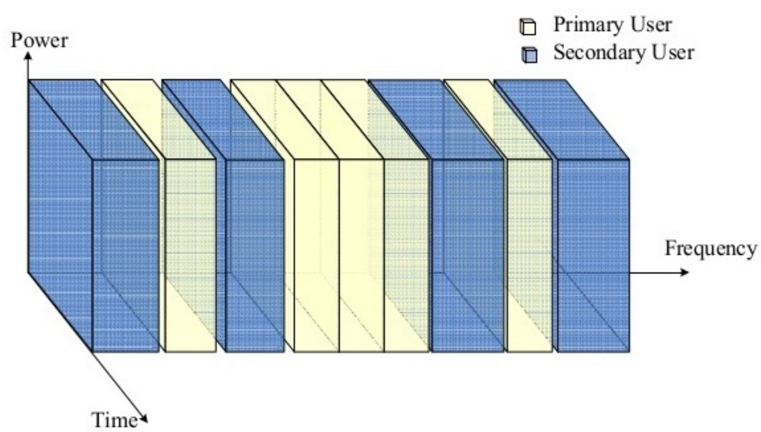


(a) Underlay.

Figure 2.9: Underlay [7]

ii. Overlay: The secondary system cooperates offering potential benefits to the primary system, for example it may aid relaying some information on behalf of the primary system, or assist detecting a hidden terminal.

iii. Interweave: SUs access PU-free channels only as shown in Figure 2.10. Such mode results in minimized interference to the PUs.



(b) Overlay.

Figure 2.10: Overlay [7]

C. How to Vacate PU Channel (Handover)

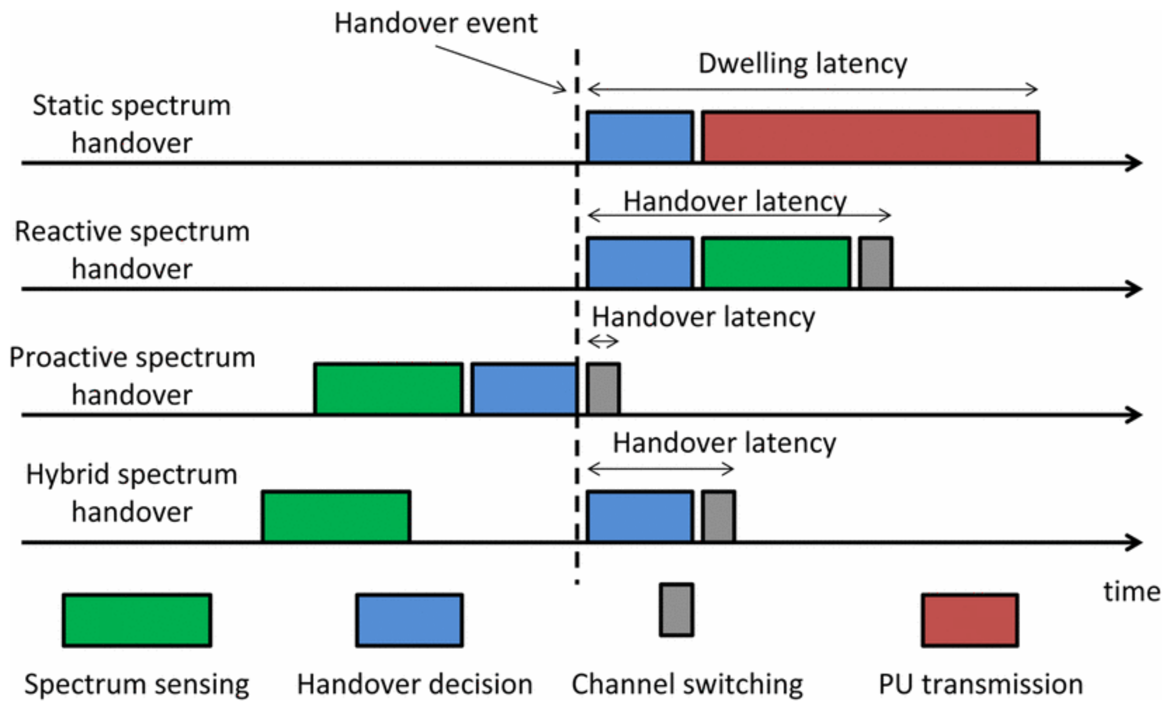


Figure 2.11: Handover [5]

- Static: The SU holds transmission and waits till the same PU channel is free again, i.e., waits till the PU finishes its whole transmission. As clear from Figure 2.11, this is the most inefficient method resulting in dwelling latency for the CR secondary system.
- Reactive: SU initiates the handover to a new channel only at PU appearance.
- Proactive: based on previously learned PU patterns, SU can predict the PU appearance moment. This results in low latency, as shown in Figure 2.11, for SU transmissions but on the cost of higher computational complexity. In case of poor PU patterns prediction, the CR system can suffer degradation.
- Hybrid: proactive to sense and reactive to decide to switch channel.

A common hardware constraint on the process of handover is: how fast can SU switch channels.

2.1.4.3 Rendezvous Techniques

A. Common Control Channel (CCC): Two SUs meet on the same channel in a cognitive radio environment with the use of dedicated common control channel (CCC). But such CCC can suffer scalability issues. It may get congested with the increase in number of users resulting in a bottleneck [10].

In addition, the availability of the CCC may change with time and if it becomes unavailable it will disrupt the SUs operation due to the loss of control messages [11]. Moreover, the CCC can suffer jamming or interference and can be viewed as a single point of failure. Any degradation on such channel for any reasons would degrade the performance of the whole CR network (secondary system) resulting in lower QoS.

One option was to have a reserved CCC channel, but then this would be against the CR concept [12]. On the other hand, if it is dynamic, it would be more difficult and nodes may lose contact with each other [13].

All the aforementioned issues have led researchers to consider different techniques, among which the *blind rendezvous* was the most practical.

B. Blind Rendezvous: Two SUs meet on the same channel without the help of any central controller nor dedicated CCC. A common approach is the use of channel-hopping (CH) where each user hops on a certain set of available channels for rendezvous with the potential neighbors [14].

The user can visit multiple channels at the same time or single channel at a time based on its radio capabilities. The more sophisticated the radio capabilities, the more the cost of the radio. Several CH algorithms have been proposed in literature for single channel visiting such as list based, random based, probabilistic based and sequence based to achieve rendezvous between CRN users [12].

In all such CH algorithms, the SU node visits channels one channel at a time, one by one in different ways based on the algorithm used. Each time the SU visits a channel, it sends a Request-To-Send (RTS) message based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in IEEE 802.11. The SU then waits on the same channel to hear the Clear-to-Send (CTS) message from its intended receiver. If it didn't hear the CTS message within the length of the time slot, it keeps hopping channels sending RTS on each new channel it lands on as shown in Figure 2.12. Therefore, the length of the time slot is equivalent to the time needed to exchange an RTS and a CTS [8].

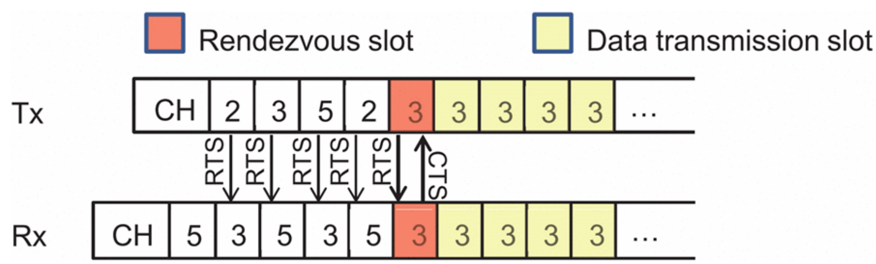


Figure 2.12: Rendezvous procedure [8]

Once the two SUs have found each other, they exchange hopping parameters and they keep hopping together. The data transmission can take place then.

Jump-Stay is the state-of-the-art algorithm that provides guaranteed blind rendezvous. It combines the common channel hopping technique with a stay stage where each user stays on a certain channel for a number of time slots [10].

2.2 Related Work

We have surveyed the available literature - as far to our knowledge - that might have touched the topic of multi-destination rendezvous. Since the closest available form communication that involves having specific multiple destinations is the multicast, multicast techniques were exploited in various technologies.

2.2.1 WiFi

With the legacy based IEEE 802.11 technology using the CSMA/CA access technique, there is no explicit mechanism for multicasting at MAC layer. Multicast packets are sent as single hop broadcast packets reaching all nodes in the neighborhood with a single transmission. The MAC layer specifications has no recovery mechanism at the MAC layer for lost broadcast frames. The use of handshaking procedures such as RTS/CTS and acknowledgments (ACKs) is not allowed. Consequently any unsuccessfully transmitted packets are lost.

This no confirmation rule was mainly to avoid CTS collisions. Researchers have suggested various techniques to solve such issue. Some suggested sending CTS at different times [15][16]. Others suggested sending the ACK following the order of the appearance of destinations in an extended multicast header or similar [17].

Another proposed scheme was using multiple unicast transmissions for multicast and multiple RTS/CTS handshakes, which proved to be reliable but incurred much delay and overhead [18]. Another proposed scheme was dividing the destinations into groups and use the group cast retries (GCR) which incurs transmitting the frame many times to each group to increase the probability of successful delivery and thus increase the reliability [19].

2.2.2 Bluetooth

In a Bluetooth scatternet, different piconets adopt different channel hopping sequences and therefore remain at different channels at the same time. When a master broadcasts a multicast message, only its slave in the piconet can receive the message. The receiving nodes will check the multicast ID of the message. If it belongs to the group, they keep the message, otherwise, the message is dropped. Forwarding nodes get to relay the message to the rest of the tree in the scatternet [20]. [21] suggests a form of collaboration between the nodes where each capable node in the path collaborates and becomes a source itself transmitting the multicast message to other nodes.

To relay multicast messages in Bluetooth through multiple unicast transmissions or flooding the scatternet piconet-wide broadcast, clearly would result in inefficiency, so a technique was proposed where the master allocates slots for slaves to communicate directly without the master interaction. The scheme can also serve as multi-slave communication hence emulating a multicast-like (i.e., group) communication within the piconet. The schedule has to be transmitted to all the slave devices so that each one of them can determine when to transmit and when to listen [22]. Other techniques involve varying the transmission power of the masters. However, that means broadcasting instead of constructing a multicast tree is still being used [23]

2.2.3 Various Wireless Technologies

While many papers [24][25][26][27][28][29] have considered the problem of multiple-destinations in the upper network layer, through routing protocols in wireless, cognitive radio networks, or considering the point of view from point to multipoint from the radio side(e.g.: directing transmissions from point (BS) to multipoint(other base stations), others were still about using the usual flooding technique in the MAC layer but in a more effective manner [30][31][32]. Other papers were concerned with multicast routing protocols and building the tree using breadth first search (BFS) in wireless multichannel mesh networks [33][34][35][36]. They relied on the assumption that the topology is somewhat considered stationary and channels available are already known to both the transmitter and receiver and devised algorithms on how to assign channels based on that; an assumption which is not valid in CR networks. In addition to the traditional idea of having multiple radios [37] which achieves better efficiency but at the hardware extra cost.

2.2.4 Cognitive Radio Networks

Multicast in CRNs followed similar techniques, one is where a single multicast transmission is broken into many small unicast transmissions introducing significant switching delay [38]. A form of collaboration is also suggested between the nodes where some of the receiving members of a multicast group assist the source by relaying the multicast message to other nodes [39] [40] [41]. Moreover for rendezvous, they relied on a common control channel (CCC) not blind rendezvous [42] [43]. Other techniques used for multicast in CRNs include optimization, machine learning and game theory [38], those need much of information - overhead - to be applicable.

Network coding (NC) [39] [38] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] has emerged as a very promising technique to enhance multicast throughput, reduce multicast time, provide protection and increase reliability in wireless networks. It is a form of source coding but is also applicable at intermediate nodes. It allows the nodes to perform packet combination instead of just forwarding them as is. The bitwise XORing of packets is considered the simplest form of network coding. It helps to increase the throughput in a network with multiple paths.

Network coding has many applications but in CRNs, it is mainly used for throughput increase. Moreover, it can also be used to create a virtual control channel which is robust against packet loss and link failure. Network coding of packets acts as if a control channel is provided as it carries all control information from nodes coded in a packet.

Considering the blind rendezvous process in the available literature, it was found that they were mainly concerned with channel hopping sequences. Some tried to manipulate the hopping sequence to achieve multiple parallel single destination rendezvous at the same time [58] while others others adopted the idea of using multiple radios to enhance the MTTR at the cost of extra hardware [59]. [10] tried to unify and shorten the channel hopping sequence for all users. It relied on spreading information between the SUs in the process of building a unified hopping sequence which would result in overhead. Moreover available channel sets of SUs dynamically change all the time, so the unified built sequence will need to be always revisited in such cognitive environment which would result in more overhead.

2.2.5 Summary of Related Work

In summary, all the available literature have not considered or have not provided any specific modification to accommodate the multi-destination rendezvous on a single radio interface in cognitive radio networks.

CHAPTER 3: PROPOSED MULTI-DESTINATION RENDEZVOUS PROTOCOL

In this chapter, we present the proposed multi-destination rendezvous protocol. First, we explain the system model considered in the proposed protocol. Then, we provide the details of the proposed multi-destination rendezvous protocol. Next we demonstrate the design considerations and the assumptions. Finally, we consider the proposed protocol within different scenarios.

3.1 System Model

In a cognitive radio network system, we have a number of SUs and a number of PUs. The PUs are licensed to use a part of the spectrum which is divided to a fixed number of channels. SUs are trying to opportunistically share those channels. The SUs can detect the PUs and other SUs within their sensing range which is governed by their hardware. Based on that sensing range, SUs can detect the vacant PU channels and therefore build their own sets of vacant channels. Once a SU has packets to transmit to another SU, it starts hopping the channels in search for that destination SU. While hopping channels, the SU sends an RTS on each channel it lands on in hopes of finding the desired destination. The system is time-slotted. Based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism in IEEE 802.11, the SU stays on each channel for one time slot to achieve a basic handshake process, it sends an RTS, then it waits on the channel for the time to receive a CTS from the desired destination, Figure 3.1, if no CTS was received, the SU just resumes channel hopping. The channel hopping sequence is generated following the state-of-the-art enhanced jump-stay rendezvous algorithm for cognitive radio networks [60]. if the SU receives the CTS from its intended destination SU, that means the handshake was successful, then both of them will stay on the same channel and data transmission can then take place between them.

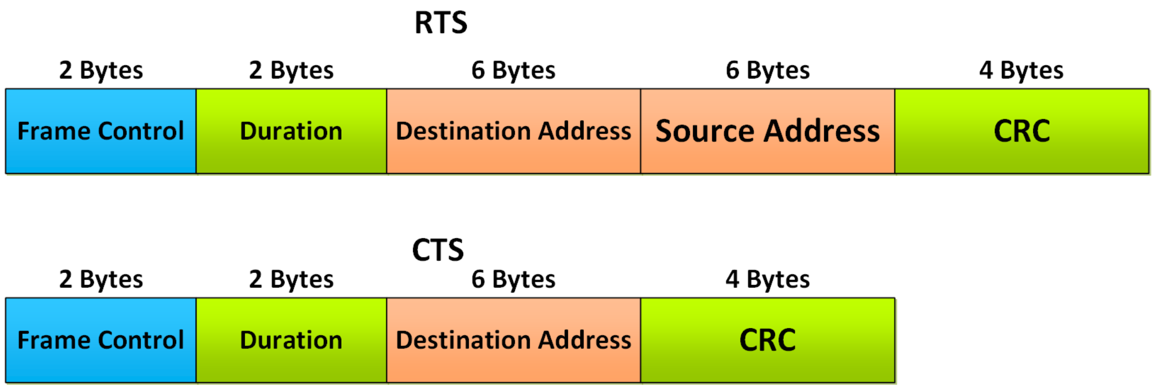


Figure 3.1: IEEE 802.11 CSMA/CA RTS and CTS

3.2 The Multi-destination Rendezvous Protocol

For the described system model in the previous section, our protocol handles the scenario where a source SU has different data packets in its buffer for multiple destination SUs. In the normal scenario, such packets will be handled with the traditional first in first served (FIFS) technique. The source SU will check the first packet in its buffer, determine a destination address and then create an RTS packet based on that destination address. The SU will then start hopping channels sending such RTS on each channel hoping to find that desired specific destination. Once the desired SU destination is found, such SU will send a CTS to complete the handshake and data transmission will take place. When the data transmission is concluded, the source SU will again consider the next packet in its buffer, check the destination address and continue to repeat the whole procedure. Such sequential FIFS procedure is inefficient specially in a cognitive environment where much of the system time would be wasted finding destination SUs and achieving rendezvous, moreover, it would result in extremely longer MTTR. Wasting the system time in achieving rendezvous instead of making use of such time to actually transmit data would result in lower throughputs. Consequently, lower throughputs in the network would mean performance degradation and lower quality of service (QoS). In a cognitive radio network where channels are not always available for SUs and where power restrictions exist on SUs' transmissions so as not to degrade PUs performance, the QoS is already lower than that of a similar primary users network. Therefore, QoS in a cognitive radio network should not be allowed to further suffocate.

Our proposed protocol comes handy to handle the multi-destination rendezvous scenario. It comprises the following procedures carried out by the source SU and the receiving SU, also interpreted in the flow charts in Figures 3.4 and 3.5 for the transmitter and the receiver SUs respectively.

1. The source SU will have the packets that need to be sent to the various destinations already available in its buffer as shown in Figure 3.2(a). The source SU will arrange and combine the packets intended for each of the destinations, as shown in Figure 3.2(b) as many as allowed by the maximum transfer unit of the system (MTU).

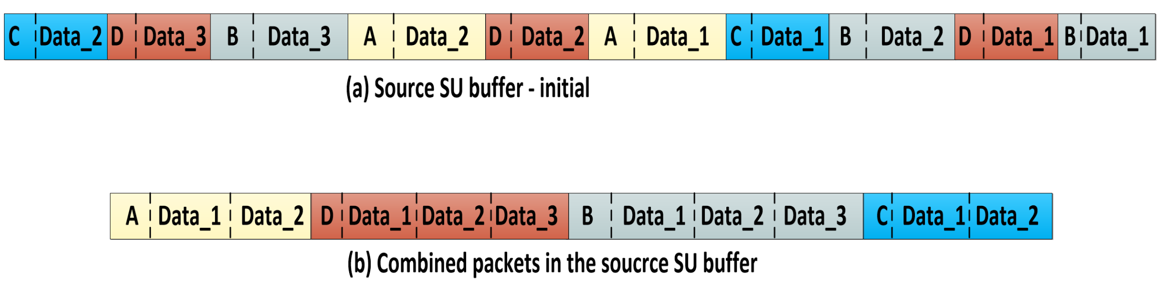


Figure 3.2: Source SU combines packets of various destinations in its buffer

2. The source SU will then apply the Network Coding technique. It will be applied on the destination addresses of the intended SUs found in the buffer. The source SU will apply the simplest form of network coding which is an XOR operation on each two destination addresses (chosen randomly from the buffer).

3.a. The source SU will create an RTS of the regular length as specified in the CSMA/CA mechanism, with the usual control frame type (01) and the subtype of 1011 indicating an RTS. Instead of inserting one destination address in the 6 bytes destination address field, it will insert the coded destination address word which now combines two destination addresses into one field (6 bytes), as shown in Figure 3.3(a).

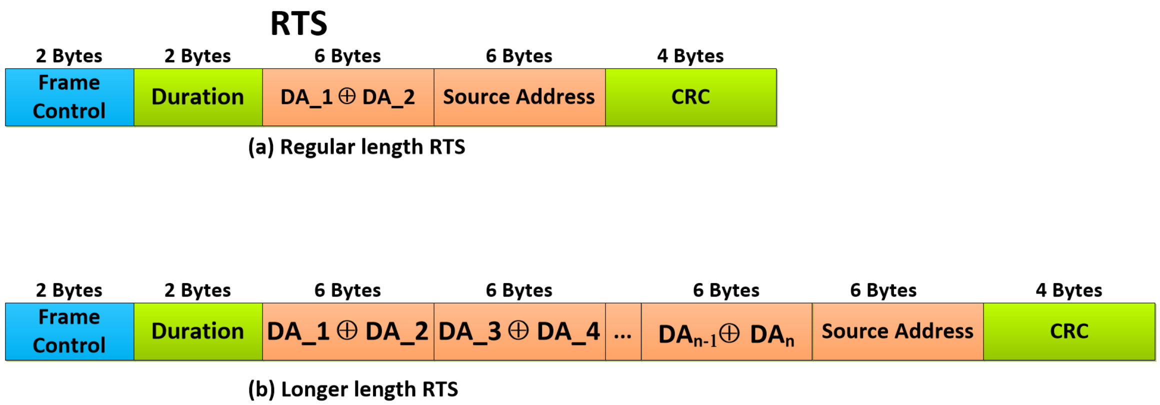


Figure 3.3: Modified RTS

3.b. Or: The source SU will create an RTS of a longer length than that specified in the CSMA/CA mechanism having multiple 6 bytes destination address fields to accommodate more destinations. The source SU will set the frame type to (11) to indicate a *modified RTS* and will set the subtype to indicate the number of destination address fields in the modified RTS to be sent.

For each 6 bytes destination address field, instead of inserting one destination address, it will insert a coded destination address word which combines two destination addresses into one field (6 bytes). It will insert the next coded destination address (the next combination of the next two destination address) in the next 6 bytes destination address field and so on till filling all the available destination address fields in the longer RTS, as shown in Figure 3.3(b).

4. The source SU will make use of the hopping sequence generated by the enhanced jump-stay rendezvous algorithm and will start hopping channels based on that sequence. The SU will send the RTS on each channel it lands on.

5. Each SU hearing the RTS on that channel will perform simple XOR processes on the destination address field(s) with its own address to decode the destination address word. Then, the receiving SU will compare the result(s) of the decode, if it matches its own address it will proceed to send a regular CTS. Otherwise it will discard the packet.

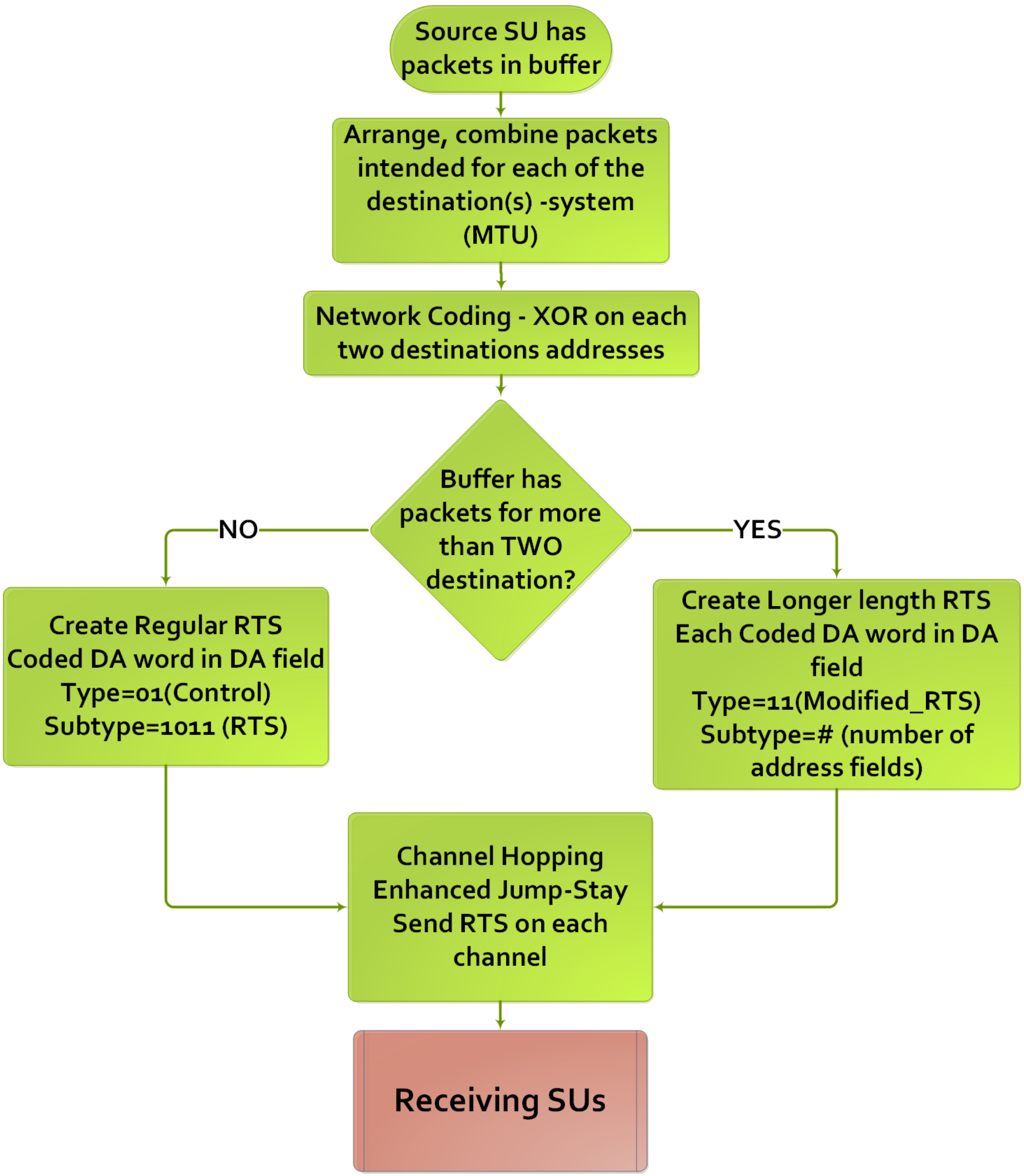


Figure 3.4: Source SU

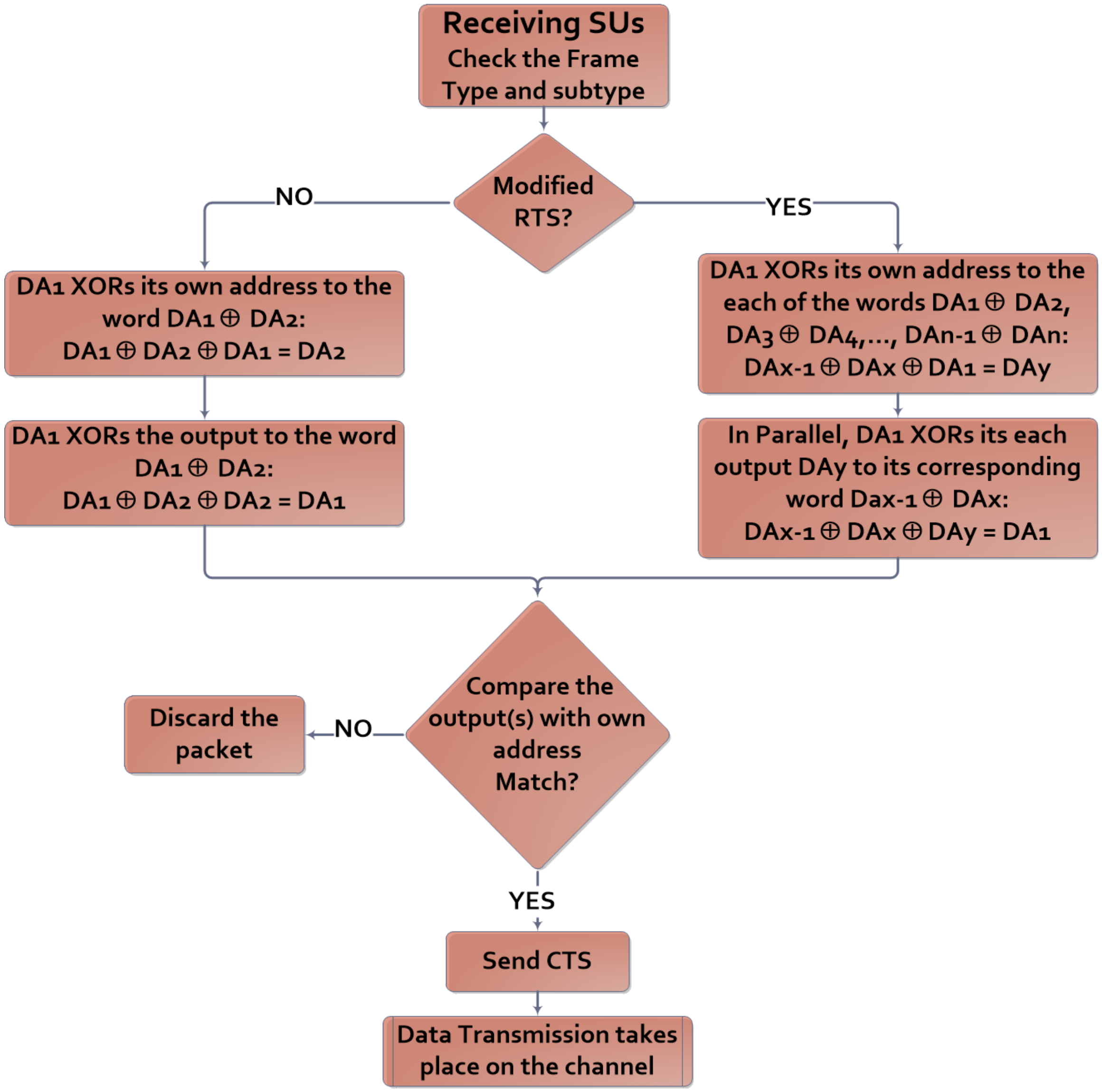


Figure 3.5: Receiver SUs

3.3 Design Considerations

3.3.1 Network Coding

We have adopted on network coding in the rendezvous process. For the first time network coding is used to code the the destination addresses of the multiple destinations, thus increase the probability of finding any one of those destinations. The simplest form of network coding which is the bitwise XORing of packets was used which can be implemented in the hardware of the Network Interface Cards (NIC) to encode and decode the addresses at both SUs. Encoding takes place at the transmitter SU while decoding takes place at the receiver. Due to the speed of performing XOR operations in the hardware, it should not be adding any significant delays to the RTS-CTS handshake process.

Based on the rules of XOR Boolean algebra where

$$A \oplus B \oplus B = A \tag{3.1}$$

$$A \oplus B \oplus A = B \tag{3.2}$$

If the RTS is received by DA_1 for example, the RTS will have the coded address word

$$DA_1 \oplus DA_2 \tag{3.3}$$

DA_1 XORs its own address to the word in (3.3) as follows

$$DA_1 \oplus DA_2 \oplus DA_1 = DA_2 \tag{3.4}$$

The output will then be XORed one more time to the coded word in (3.3) as follows

$$DA_1 \oplus DA_2 \oplus DA_2 = DA_1 \tag{3.5}$$

As per (3.5), the resulting output of the process is matching DA_1's own address, then that receiving SU will send a CTS.

Otherwise, if the received RTS does not contain DA_1, the following will take place

$$DA_3 \oplus DA_4 \oplus DA_1 = X \tag{3.6}$$

and then

$$DA_3 \oplus DA_4 \oplus X = Y \tag{3.7}$$

As seen from (3.7) $Y \neq DA_1$. Therefore, in such case, the packet will be discarded. Such operations can be performed in parallel in the hardware on all the destination address fields.

3.3.2 New RTS Format

The regular length as specified in the 802.11 CSMA/CA, is 20 bytes in length and has a control(01) frame type and the subtype of 1011 indicating RTS as per the standard as shown in Figure 3.6.

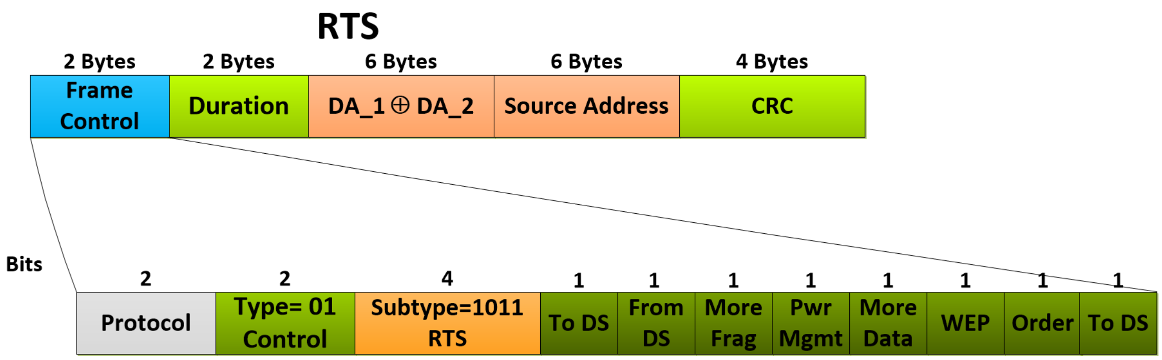


Figure 3.6: Regular length RTS Frame Control

For the RTS to be able to adapt to the presence of multiple-destinations, it has been modified. A new RTS has been designed with variable length. The new *modified RTS* will have multiple destination address fields and the frame type for such modified RTS would be (11). In the frame control part of the RTS, the subtype of four bits would be used to indicate the number of destination address fields. With the subtype field of size of four bits, that means it can represent numbers from 0 to 15. The number will indicate the number of destination address fields in addition to the original existing destination address field of the original regular RTS. A subtype value of zero will be used to indicate the value of 16 destination address fields. That implies that the modified RTS can have from 2 to 17 destination address fields, supporting from 4 to 34 coded destination addresses. The RTS would be dynamic, i.e., its length will vary based on the number of destinations the source has in its buffer. The relation between the subtype value and the number of destination addresses n can be obtained using formula (3.8), also shown in Figure 3.7.

$$Subtype_b = (n-1)_b/2_b \tag{3.8}$$

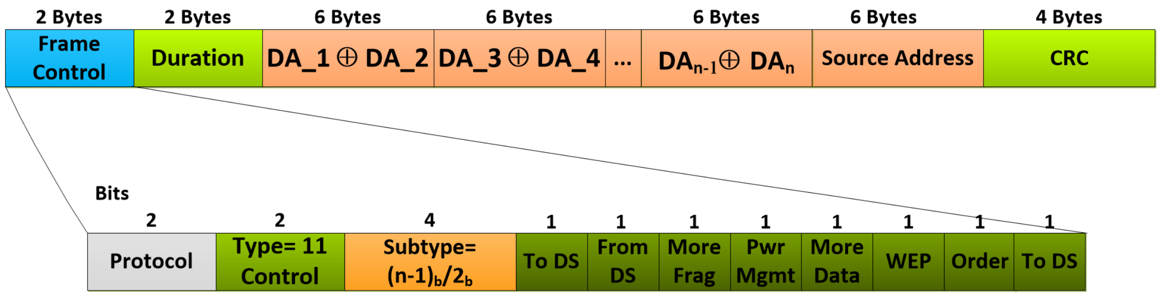


Figure 3.7: Longer length RTS Frame Control

For example, a source with packets to 5 destinations in its buffer will form a modified RTS having 3 destination address fields and will set the subtype to 2 $(0010)_b$, A source SU with packets for 8 destinations in its buffer will form an RTS having 4 destination address fields and will set the subtype to 3 $(0011)_b$, the type will be set to (11) for both examples.

The subtype field size will not be changed from the original standard and will be kept at 4 bits to allow for some backward compatibility although this will impose the limit on the maximum number of addresses that the RTS can hold up to 34.

Each SU receiving an RTS will need to check the frame type and the subtype field to be able to determine and endorse the length of the received RTS, whether it is the regular RTS or the modified one and how many destination address fields are included in this modified RTS.

3.3.3 Time Slot and the Length of RTS

As previously mentioned in section 3.1, the system is time-slotted. The SU stays on each channel for one time slot to achieve a basic handshake process, after it sends an RTS, it waits on the channel for the time to receive a CTS from the desired destination, if no CTS is received, the SU just resumes channel hopping. Consequently we can deduce that the minimum length of the time slot is equivalent to the time needed to exchange the RTS and CTS messages. Depending on the bandwidth of the channel, the minimum time slot is given by equation 3.9

$$t_{min} = \frac{(RTS + CTS)bits}{Bandwidth} \tag{3.9}$$

Of course the length of the time slot should accommodate for round trip delay, propagation and processing delays and synchronization mismatches. [61] discusses the scenario where the system is asynchronous.

We deduced that the length of the time slot is highly correlated to the length of the RTS. The minimum length of the RTS is the length of the regular RTS which can accommodate at least one destination address and with the use of network coding through our protocol, it can accommodate two. But when we come to discuss the optimum length of the RTS, we have to consider some factors. For instance, the length of the RTS cannot increase indefinitely, it is governed by the subtype field size and can hold a maximum of 34 destination addresses in 17 destination address fields.

Although the length of the RTS will not affect the normalized throughput, it will affect the MTTR. In terms of time slot count, the longer the RTS, the less the MTTR in number of counted time slots but when we convert that into time, it means that as the length of the RTS increases, the MTTR will increase as well resulting in longer delay.

There is a trade-off here, although handling multiple-destinations together with the modified longer RTS would result in noticeable enhancement in performance and network utilization, at a certain length of the longer RTS, the resulting rendezvous delay may become even longer than the original MTTR of the sequential technique of handling multiple destinations. Moreover, increasing the length of the RTS, hence the system time slot means the packets would be more susceptible to more bit error rate (BER) and would be liable to fragmentation at lower network layers. In addition, under the same packet arrival rate, a longer time slot would result in a higher utilization among the network channels, that is because more arrivals will occur in the same time slot (as it is longer in time) and such arrivals will not be served in a timely manner, thus causing channels' overcrowding and eventually blockings in the next time slots. Consequently, care must be taken when choosing the length of the RTS in order to keep the MTTR and the aforementioned factors optimum. However if the system time slot is already long and can support a longer RTS, then the RTS length can be increased to the upper bound such that the length of the time slot is slightly larger than t_{min} as per equation(3.9).

3.3.4 Channel Hopping Sequence

Our proposed protocol is not restricted to a specific channel hopping sequence. Many papers have researched techniques to generate the channel hopping sequence [10] [11] [12] [14] [62] [63] [64] [65] [66] [67], we have decided to adopt the enhanced jump-stay rendezvous algorithm for cognitive radio networks[60]. Therefore, the channel hopping sequence is generated following the state-of-the-art enhanced jump-stay rendezvous algorithm for cognitive radio networks. The generated sequence is based on four parameters which are the total number of channels of that part of the spectrum, the smallest prime number greater than the total number of channels (P), the available channel set for that SU and a trial – time slot – counter. Of course the enhanced jump-stay rendezvous algorithm guarantees rendezvous without the need of time synchronization. It also provides an upper-bound on the maximum time to rendezvous and the MTTR in the order of $O(P^2)$. In the enhanced jump-stay rendezvous algorithm each round consists of a jump-pattern and a stay-pattern. The previously mentioned parameters are used to generate both patterns. The length of the jump-pattern is *3P* time slots followed by a stay-pattern that lasts for P time slots. The user starts with index i and keeps hopping in [1,P] with step-length r – which is an integer in [1,M] – using module operations on P. The index i is switched to the next number every round of $4P$ time slots using round-robin. In the stay-pattern, the user stays on channel r.

3.4 Design Assumptions

The following assumptions were taken into consideration while designing the protocol. First, channel reuse does not occur except at further distances and since our protocol is concerned with multiple-destinations for the same source SU, all destinations, if available, are assumed to be within one single hop reach from the source SU. In addition, neither PUs or SUs will reuse the same channel within the vicinity of a single hop.

Second, the whole concept of CR networks is that SUs can opportunistically share the radio resources that might have equal or unequal access rights. In cognitive radio SUs are allowed to transmit as long as they do not degrade PUs' communication, also the radio resources should be already satisfying all the needs of the PUs and exceeding their requirements. PUs should not be suffering any blockings. Otherwise, if the resources are not enough for PUs, then there is no point of having a cognitive network of SUs who are trying to share what is already insufficient for the PUs. In other words, it is expected that the network utilization is not fully saturated.

Third, since all the SUs are in the vicinity of one hop of each other, it is expected that there should be some similarity in the available channels observed by SUs. For two SUs to be able to achieve rendezvous, both of their observed available channel sets should have at least one common available channel, their channel sets do not have to be identical though. Otherwise if all the channel sets have nothing in common, it would be impossible for the SUs to communicate.

Fourth, as long as the source SU does not have packets for multiple destinations in its buffer, it will continue to treat each individually arriving packet in the traditional first-in-first-out way. As long as the buffer has not more than one packet, the SU will proceed to send this packet immediately without delay or waiting to form a group of multiple destinations.

Fifth, interference from PU will not be considered, we assumed that a PU will not suddenly come back to a used channel, i.e. it won't interrupt an ongoing SU transmission. To consider such case, means that the SUs will need to vacate the channel and move to another channel in order to continue their transmission but such channel handover by SUs is out of the scope of our work.

3.5 Protocol Behavior in Various Scenarios

Having a protocol that can perform rendezvous for multiple destinations at the same time means additional scenarios would exist that were not there with the case of sequential rendezvous of a destination by destination.

3.5.1 Collision Among Multiple Destinations During Rendezvous

The source SUs has packets to multiple destinations and has applied the network coding and started the channel hopping following the enhanced jump-stay algorithm. Destination SUs as well follow the enhanced jump-stay algorithm to generate their channel hopping sequences. The enhanced jump-stay algorithm may generate the same next channel number for more than one SU at the same time.

If the source and one destination are within those SUs with the same generated channel number, rendezvous will occur with no problems as any other SUs getting the RTS message will try to decode the destination addresses field but will not find their own address in the result of the decode process and therefore will just discard the RTS message. If the source alone is among those SUs while no intended destination SU got the same channel number then no rendezvous will occur and still no problems will happen, any other non-intended SU on the same channel will just discard the RTS message.

It may happen by chance that the enhanced jump-stay algorithm generates the same next channel number for the source SU and more than one SU of the source's intended destinations. In such case, each intended destination SU on the channel will get the RTS message, decode the destination addresses field and once it finds itself within the intended destinations, it will proceed to send a CTS to declare its presence on the channel and complete the handshake. In a cognitive environment, the SUs are blind, i.e., they do not have information about the presence of other SUs in their vicinity nor on the same channel. Consequently in such scenario where each intended destination proceeds to send a CTS, a collision will occur leading to fail the handshake and therefore the rendezvous at such time slot.

Although such wasted time slot would contribute towards a longer time to rendezvous, the probability of occurrence of such scenario is very low (as shown in the simulation results in Chapter 4) as it happens as a result of intersection of all the previously mentioned conditions in the scenario occurring all together at the same time.

3.5.2 Post First and Second Rendezvous

As mentioned in section 3.3.2, the number of additional destination address fields in the RTS is governed by the number of destinations addresses through the formula in (3.8). If the number of the destinations is odd, then the last address field in the RTS will have only one address, as shown in Figure 3.8.

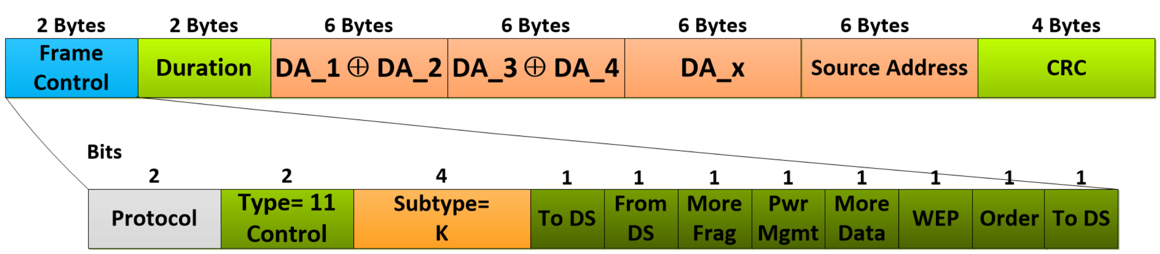


Figure 3.8: The last address field has one destination

As per equation (3.3), the coded word in that last destination address field will be

DA_x as shown below

$$DA_{x-1} \oplus DA_x = 0 \oplus DA_x = DA_x \tag{3.10}$$

The source SU has packets for multiple-destinations and has started hopping channels following the enhanced jump-stay algorithm looking for any of its destinations. Once a destination is found, they remain on the channel and exchange data. Assuming the scenario that that first destination found was the one and only address in the last destination address field, DA_x, so the source SU will remain on the channel with DA_x, transmit the data packets to DA_x and then remove DA_x from the destination list being pursued. The source SU will also remove the DA_x from the destination address fields. Since DA_x was the only address in that last destination address field, that means the source SU will remove that whole destination address field as it will be no more needed and thus the length of the RTS will decrease, also the subtype field in the RTS frame control will be decremented by 1. This is shown in Figure 3.9.

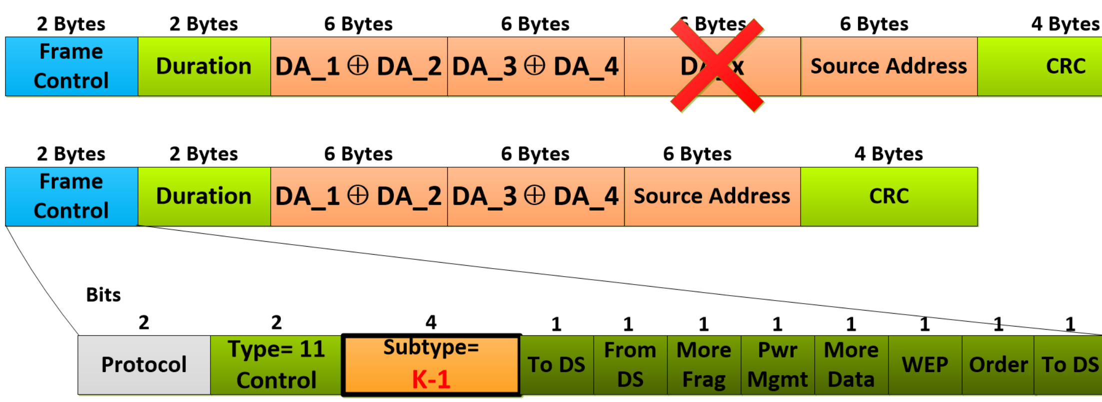


Figure 3.9: The last address field is removed and the subtype is decremented

If the number of the destinations is even, then the each address field in the RTS will have a coded word of two addresses. the source SU would be looking for its destinations, and will find them and then transmit data to them one by one. Considering only one of those destination address fields, say the source SU found the first destination in that destination address field, it will proceed to transmit data to it and then remove its destination address from that destination address field leaving only one destination address. The subtype value would not be changed yet. Such process is shown in Figure 3.10.

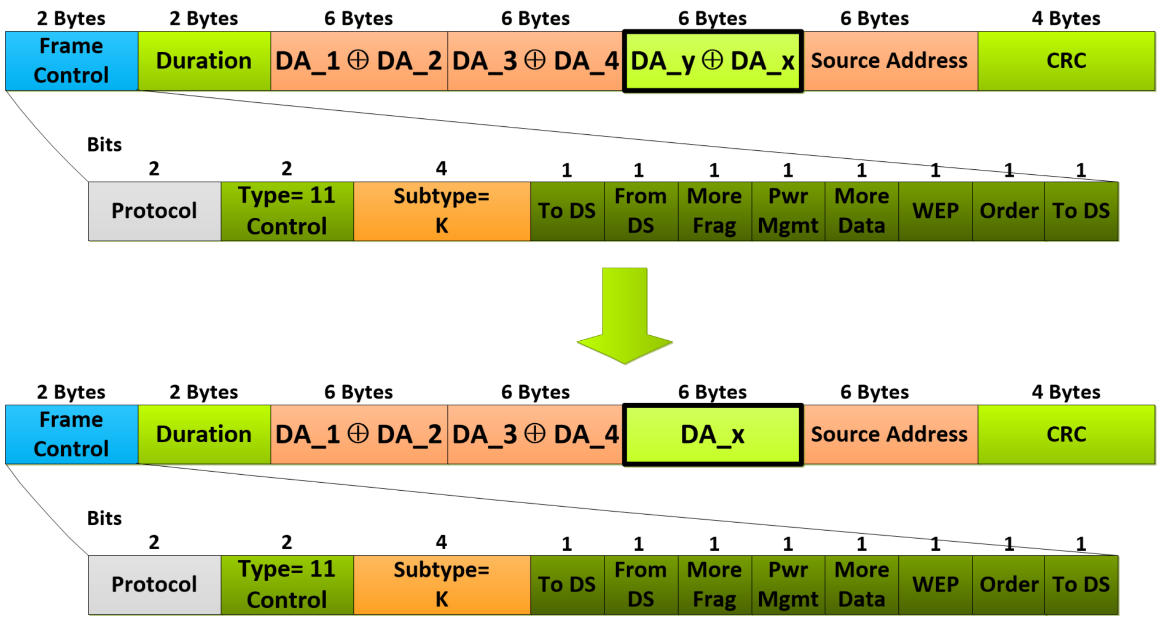


Figure 3.10: Even number of destinations

After that, the case would be similar to that of the odd number of destinations mentioned earlier in this section and after the rendezvous and data transmission to the second destination address in that destination address field, similarly, the source SU will remove that whole destination address field as it will be no more needed, the length of the RTS will decrease, also the subtype field in the RTS will be decremented by 1.

The above two scenarios will continue to repeat throughout the whole process till finding all the destinations and will apply on all address fields wherever their location within the RTS.

3.5.3 Repeated Collision Among Multiple Destinations During Rendezvous

As mentioned earlier in section 3.5.1 and due to the blindness of SUs about the presence of each other in such a cognitive environment, if the source SU and more than one SU of that source's intended destinations are on the same channel, each intended destination will proceed to send a CTS and a collision will occur leading to fail the handshake and therefore the rendezvous at such time slot.

A coincidence may occur that the randomly generated parameters *i, r and t* used with the Enhanced Jump Stay function, that is the function that generates the hopping sequence *EJSH(i,r,P,M,t)* [60], are the same for two of the intended destinations, so these two keep colliding, not just for once but each time because their generated channel numbers will always be the same as the parameters used to generate the channel numbers are similar. Consequently, failed rendezvous will be counted each time as successful rendezvous is never achieved due to repetitive collisions which means that their MTTR would go to infinity.

Although the probability of occurrence of such issue is very low (as deduced from the simulation results - Chapter 4), the severity of such issue is that it drives the MTTR to infinity and therefore causes severe network performance degradation thus imposing the must to find a solution for such severe issue.

To overcome such issue, each destination can count its collisions, that is to count the number of RTSs received from that same source SU that it actually replied to with a CTS but never got data packets after that from that specific source. If its collision counter exceeds a certain number X (ex: 10), then it randomly re-chooses a parameter r again which will be used later on by the function *EJSH* to generate the hopping sequence

CHAPTER 4: PERFORMANCE EVALUATION

In the previous chapter, we have proposed a new protocol to handle the scenario of a source SU having packets to send for various destinations in a cognitive radio environment over a single radio in an effective efficient way. In this chapter we do the performance evaluation to our multi-destination rendezvous protocol. We evaluate our protocol performance to demonstrate its effectiveness and to show that it provides the mentioned performance enhancement. The performance evaluation provided in this chapter is threefold. First we evaluate our protocol and monitor the enhancement in the MTTR. Then we observe a downside of having multiple-destination rendezvous and analyze the occurrence of that shortcoming. Last, we consider the effect of our protocol on one of the most important key performance indicators of a network which is the throughput.

4.1 Rendezvous Delay

We started our evaluation by analyzing the MTTR. The MTTR is a key performance indicator in any cognitive radio network and such analysis will determine the effectiveness of our proposed protocol

4.1.1 Effect of Number of Destinations and Number of PUs

Simulations presented in this section were carried out at a time slot length of 480 bits, i.e. 60 bytes, over a link speed which was assumed to be 2 Mbps. Since the length of the original CSMA/CA RTS + CTS together is 34 bytes, that means such time slot allows for additional 26 bytes. Consequently the RTS can accommodate up to 10 destination addresses. The number of SUs used in simulations is 225. The number of PUs vary within the simulation between zero and 150. Moreover, both PUs and SUs are randomly distributed within the simulation area of 12 units. While the sensing range for an SU is assumed to be 2.2 units, but the range where SU signals have enough power to actually achieve a rendezvous and transmit data - not just sense the presence of another radio - is limited to 1.2 units. The number of channels of the primary system is set to 100. The packet arrival rate for both the PUs and the SUs follows a Poisson process with mean arrival rate of 50 packets/sec for both. The size of transmissions for each of the PUs and the SUs is assumed to be consuming 50 time slots and 20 time slots respectively, the size is fixed throughout the simulations to keep the comparison. The size of the SU transmissions was set to be smaller than the PU transmissions' size because at such cognitive environment, the PU can appear at the channel at any moment, so smaller sizes of transmissions would intuitively have higher probability of being successfully transmitted without being interrupted by the returning PU. The exact optimum size of the SU transmissions is out of the scope of our work, more on the optimum packet size can be found in [68].

Figure 4.1 shows the percentage of enhancement in the MTTR with the rendezvous of varying number of destinations. Our proposed protocol provides a significant enhancement over the traditional sequential technique. The enhancement increases with the increase in number of destinations undergoing trials to rendezvous simultaneously by the source SU. The percentage of enhancement varies from 20% when considering two destinations and goes up to 70% when considering an RTS with 10 simultaneous destination addresses.

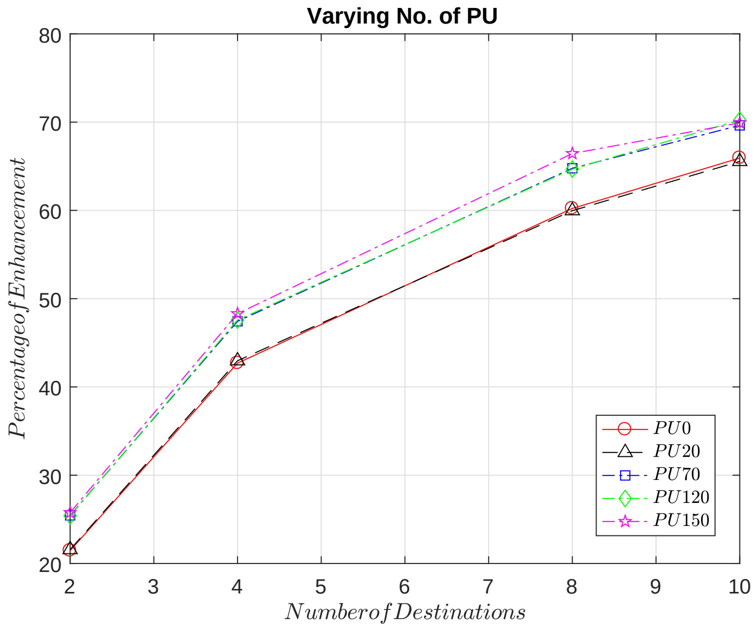


Figure 4.1: Percentage of enhancement of MTTR with the rendezvous of varying number of destinations

In Figure 4.2, we show the effect of varying number of present PUs in the system on the percentage of enhancement of MTTR. Considering the case of 4 destinations rendezvous, the enhancement varies from approximately 43% to 48% with the increase of number of PUs from zero - i.e. no PUs using the system channels at all - to 150 PU utilizing the system channels. Same effect goes for different cases when considering different number of destinations for multi-destination rendezvous as clear from Figure 4.3. The lowest percentage of enhancement in MTTR is recorded when the PUs are completely absent from the system while the maximum recorded percentage is that when the number of PUs in the system peaks to 150.

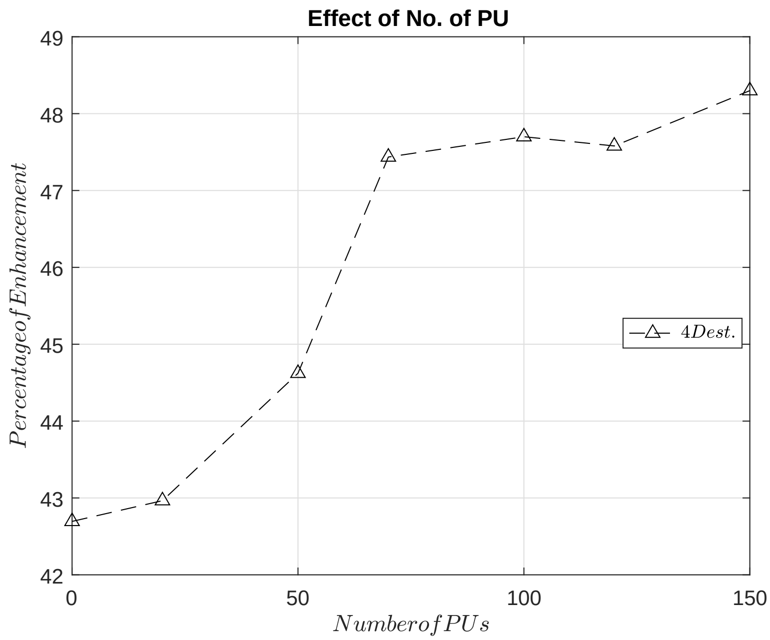


Figure 4.2: Percentage of enhancement of MTTR with the rendezvous of four destinations for varying number of PUs

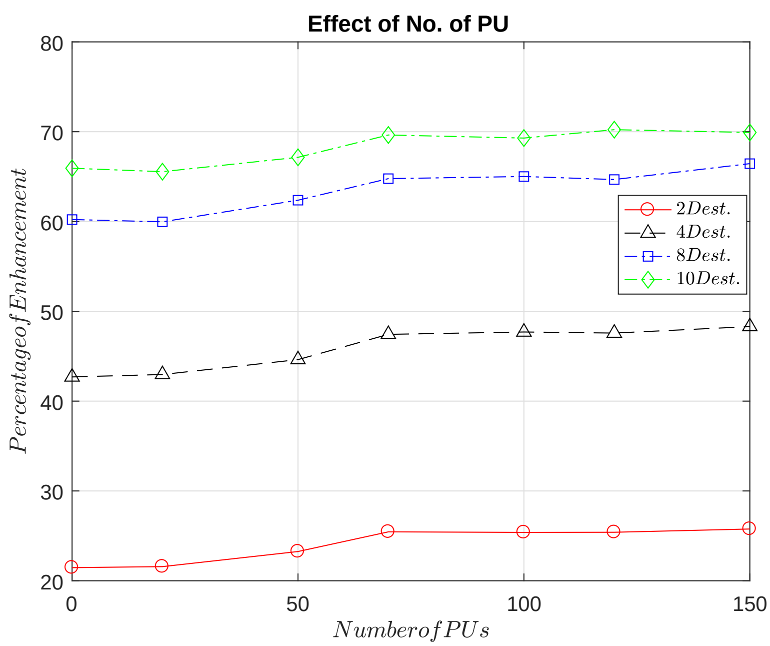


Figure 4.3: Percentage of enhancement of MTTR with the rendezvous of varying number of destinations for varying number of PUs

Such trend of increase in enhancement with the increase in number of PUs may seem extraordinary and against the expectations where usually the increase in number of PUs in a cognitive radio network is expected to affect the SUs negatively. That can be explained as follows, the number of channels in the system are fixed and each SU can sense and identify the vacant portion of those channels based on the PU activity in the system. A certain SU having traffic for multiple destinations would be hopping all over its set of available channels looking for those N destinations. As the number of PUs in the system increase, the number of available channels for SU use would decrease, which in turn means the set of available channels for the SU would become smaller. Consequently the SU would need to hop on a smaller number of available channels in its set to find the same group of N destinations and therefore the probability of eventually finding the whole same group of N within a smaller number of channels would increase, thus leading to a *relatively* shorter MTTR compared to the same group of N when utilizing the traditional technique and as a result a better percentage of enhancement for the MTTR. So we can deduce that our protocol exploits the problem of lack of resources and transforms it to an advantage to the SUs to benefit from.

In the extreme case where PUs are using all the channels all the time, there will be no vacant channels for SU use. Moreover, SU channel sets would always be empty and with such empty channel sets, SUs cannot start hopping and therefore no rendezvous can be achieved. Consequently SUs cannot communicate in such extreme case.

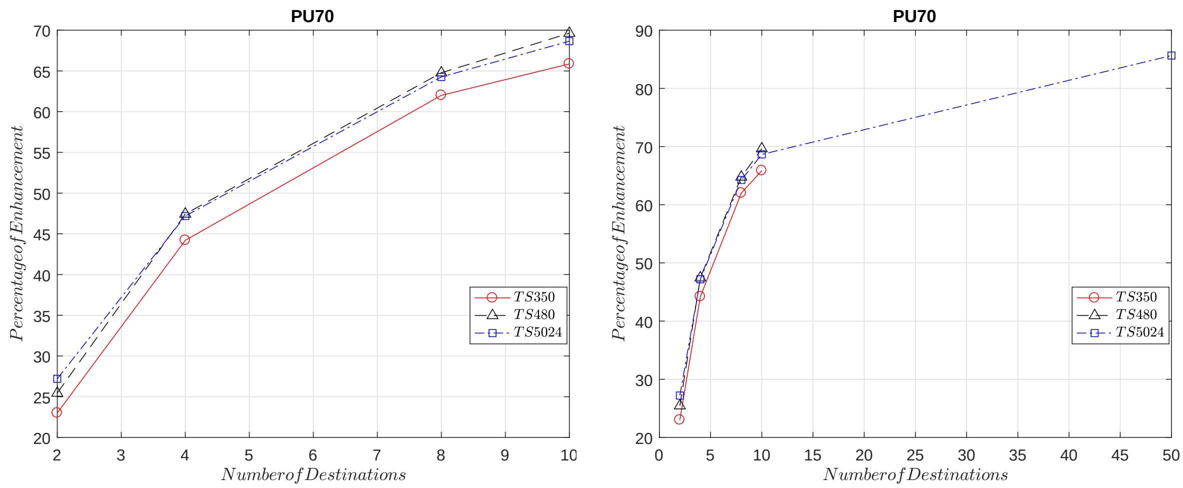
4.1.2 Effect of Time Slot Length

As mentioned earlier in section 3.3.3, when it comes to selecting a length for the system time slot, there is a trade-off to be considered, the enhancement in performance maybe coupled with a longer MTTR among other factors which include BER, fragmentation and blockings. So we have carried out the experiments in the current section in order to analyze the effect of the time slot length on the MTTR. Same simulation parameters apply from the previous section, the difference is that we vary the length of the time slot. In order to study the effect of the time slot length thoroughly, we vary the length from 350 bits going up to an extreme length of 5024 bits which can handle extreme values when it comes to the number of destination addresses that can be included in the RTS. In addition, for such case of the longer time slot, we consider the case of 50 concurrent destinations as an extreme number of concurrent destinations in the simulation. Although the limit imposed by the subtype field allows the RTS to handle only up to 34 destinations, we have simulated for up to 50 to provide a further sense of the trend of enhancement by our proposed protocol and because future designs of the RTS may provide a technique to overcome this limitation and include more destination addresses and in such case, our protocol would still be applicable and able to provide enhancement over traditional techniques. Moreover, for the time slot of 5024 bits and with such extreme length of the time slot, the arrival rate would cause a vast number of arrivals in such long one time slot leading to much higher utilization of the system channels and therefore blockings. We had to decrease the arrival rate to allow for same network utilization levels for all time slot values to keep the comparison fair between them.

We started by simulating the performance at the presence of an average number of PUs in the system which is 70. In Figure 4.4(a), the percentage of enhancement of MTTR is plotted against varying number of destinations for three different time slot lengths, 350 bits, 480 bits and the longer time slot of 5024 bits. It is obvious that the length of the time slot has very minor effect on the enhancement percentage, for example at 2 destinations, the percentage varies from 23% to 27% when the time slot length varies widely between 350 and 5024 bits (more than 10 times), similarly at the 4 destinations, it varies from 62% to 64%.

In Figure 4.4(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, our proposed protocol shows great improvement over the traditional technique, the enhancement percentage continues to increase with the increase in number of concurrent destinations till it reaches more than 80% at the extreme case of 50 destinations.

Hence we can deduce, as long as the system is time-slotted, the percentage of enhancement of MTTR would be time slot length independent.

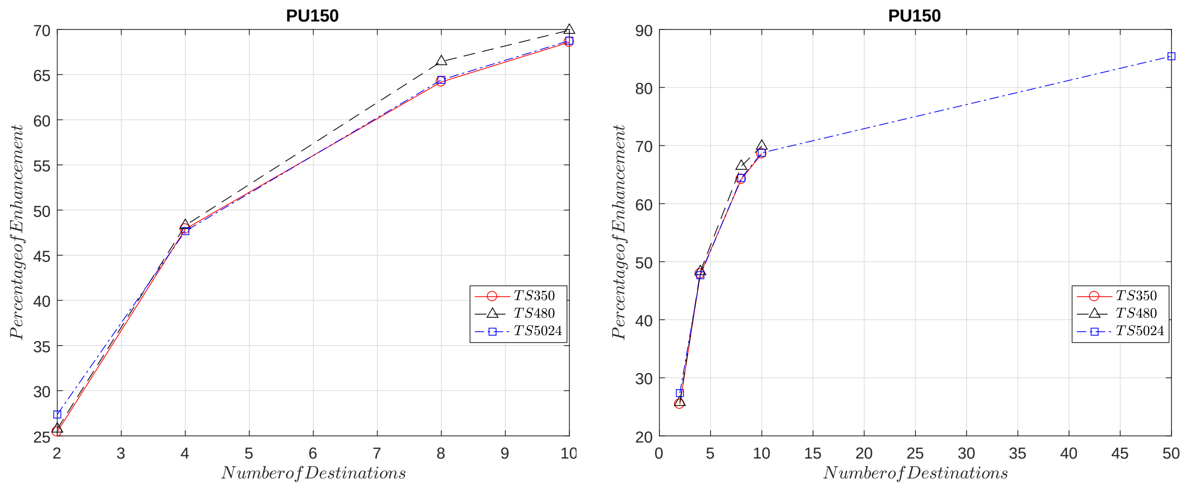


(a) Up to 10 destinations

(b) up to 50 destinations

Figure 4.4: Percentage of enhancement of MTTR with the rendezvous of varying number of destinations taken at different time slot lengths and at 70 PUs

To further confirm that position, we also simulated the performance at the presence of a higher number of PUs in the system which is at 150 PUs.



(a) Up to 10 destinations

(b) up to 50 destinations

Figure 4.5: Percentage of enhancement of MTTR with the rendezvous of varying number of destinations taken at different time slot lengths and at 150 PUs

In Figure 4.5(a), similar to the previous figure, the percentage of enhancement of MTTR is plotted against varying number of destinations for the three different time slot lengths. At 2 destinations, the percentage varies from 25% to 27% when the time slot length varies between 350 and 5024 bits , and at 4 destinations, it is approximately the same at 47%. Moreover, the percentage of enhancement is almost approximately the same for the rest of varying number of destinations over the different time slot lengths.

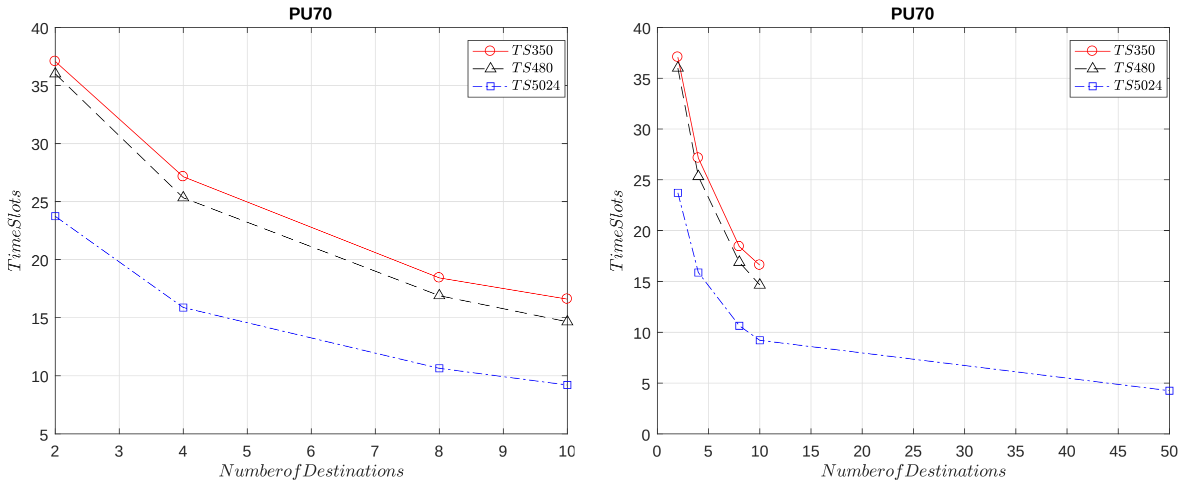
In Figure 4.5(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, our proposed protocol shows the improvement over the traditional technique here as well, the enhancement percentage continues to increase with the increase of number of concurrent destinations till it reaches more than 80% at the extreme case of 50 destinations.

4.1.3 Time To Rendezvous

In this section we focus on the time to rendezvous rather than the performance enhancement. Results presented in this section were simulated on similar parameters to those used in section 4.1.2 and were performed for the same lengths of time slots, 350, 480 and the extreme case of 5024 bits. Similar to section 4.1.2, simulations were carried out at the presence of an average number of PUs in the system which is 70.

In Figure 4.6(a), the MTTR, in number of time slots, is plotted against varying number of destinations for three different time slot lengths, 350 bits, 480 bits and the longer time slot of 5024 bits. When comparing the MTTR for the same number of concurrent destinations at different time slots, it is obvious that as the length of the time slot increases, the number of time slots needed to achieve rendezvous decreases. For example at 2 destinations, the MTTR is 37 time slots at the length of 350 bits, which decreases to 36 time slots when considering the longer time slot of 480 bits and abruptly drops to 23 when considering the longer time slot of 5024 bits. Similarly at the 4 destinations, the MTTR is 27 time slots at the length of 350 bits, which decreases to 25 time slots when considering the longest time slot of 480 bits and abruptly drops to 15 when considering the longest time slot of 5024 bits.

In Figure 4.6(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, the MTTR continues to decrease with the increase of number of concurrent destinations till it reaches a low value of 4 time slots at the extreme case of 50 destinations.

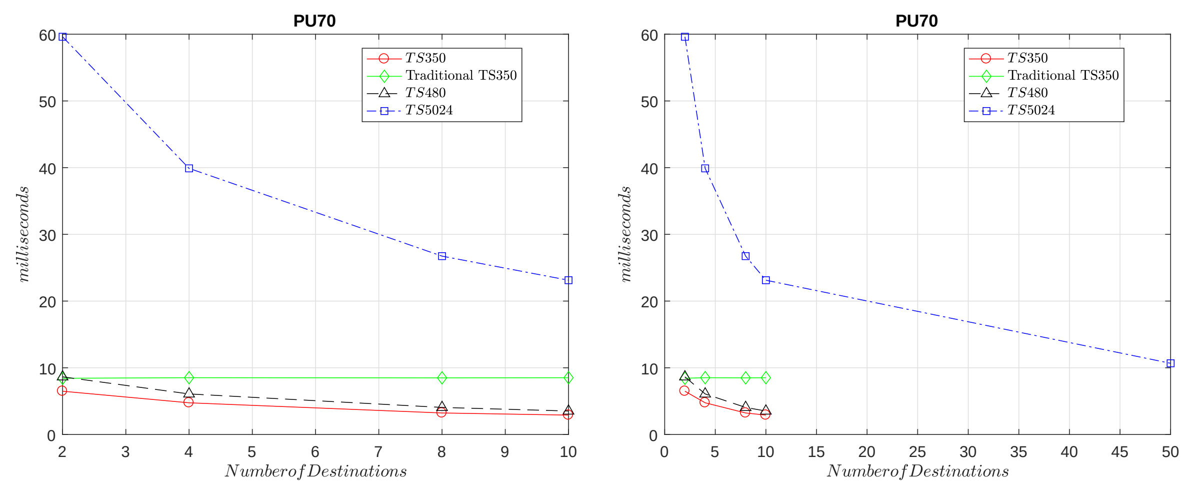


(a) Up to 10 destinations

(b) up to 50 destinations

Figure 4.6: The MTTR in number of time slots, with the rendezvous of varying number of destinations for different system time slots

To further analyze the interpretations of the outputs from the previous results, we convert the MTTR to time in milliseconds and re-observe the outputs in Figure 4.7. In Figure 4.7(a), we plotted the outputs by our proposed protocol for time slot lengths of 350, 480 and 5024 bits while the diamond solid line represents the traditional technique at the shortest time slot of 350 bits. As observed from the figure, the outputs by our protocol provide shorter MTTR compared to the traditional technique at the shorter time slot length except for the time slot length of 5024 bits. We have already observed the percentages of enhancement per MTTR for each time slot length in the previous section, but here we compare our outputs to the *shortest* achieved time by the traditional technique to tackle the topic of the optimum system time slot length from one aspect which is the delay till rendezvous. We can observe the trade-off mentioned in section 3.3.3, using the traditional technique at a time slot length of 350 bits yields an MTTR of approximately 8.5 milliseconds which is of course irrelevant to any number of destinations the source SU might need to send to, but with our proposed protocol, this MTTR decreases to 6.5 ms at the same time slot length considering 2 concurrent destinations and 4.8 ms when considering 4 destinations. Moreover, if we decide to increase the system time slot length to accommodate up to 10 destinations, that is a length of 480 bits, then the MTTR goes to 8.6 ms (1% variation) but then decreases to 3.5 ms when considering 10 concurrent destinations. On the contrary, if we decide to increase the system time slot length to 5024 bits accommodate an extreme number of concurrent destinations, then the MTTR increases dramatically to 60 ms when handling 2 destinations and then it decreases as the number of concurrent destinations handled increase till it becomes as low as 10 ms at the extreme case 50 concurrent destinations as observed from Figure 4.7(b).



(a) Up to 10 destinations (b) up to 50 destinations

Figure 4.7: The MTTR in milliseconds, with the rendezvous of varying number of destinations for different system time slot lengths

We also provide Figure 4.8 for further reference and observation where all solid lines represent MTTR in milliseconds by the traditional technique versus those by our proposed protocol, the percentages of enhancement in MTTR were previously provided in section 4.1.2 in Figure 4.4

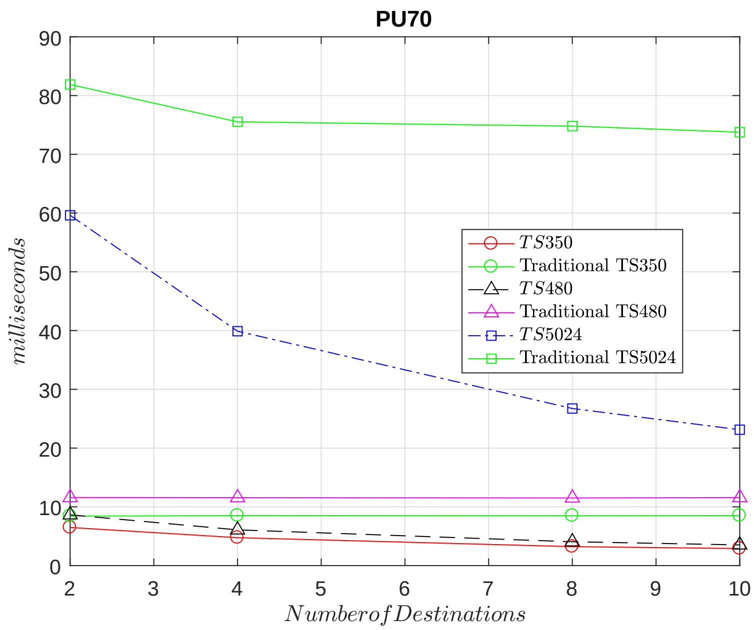


Figure 4.8: The MTTR in milliseconds, with the rendezvous of varying number of destinations for different system time slot lengths Versus Traditional

4.2 Collisions

In section 4.1, we have showed the enhancements in the delay till rendezvous associated with the utilization of our protocol. In this section, we show the effect of an undesirable aspect associated with introducing the multi-destination feature of our protocol which is collisions. As mentioned earlier in section 3.5.1 and due to the blindness of SUs about the presence of each other in such a cognitive environment, if the SU and more than one of the source's intended SU destinations are on the same channel, each intended destination will proceed to send a CTS and a collision will occur leading to failure of the handshake and therefore the rendezvous at such time slot.

4.2.1 Effect of Number of Destinations and Number of PUs

Simulations presented in this section were carried out at the same simulation environment and using the same parameters as those of section 4.1.1 except for the number of SU which was increased to 2025, also the number of PUs vary within the simulation between zero and 150.

Figure 4.9 shows the percentage of time slots where collisions have occurred during the cognitive radio network system simulation varying with the rendezvous of different numbers of destinations. The percentage of time slots with collisions increases with the increase in number of destinations undergoing trials to rendezvous simultaneously by the source SU. The percentage varies from 0.01% when considering two destinations and goes up to 0.33% when considering an RTS with 10 simultaneous destination addresses.

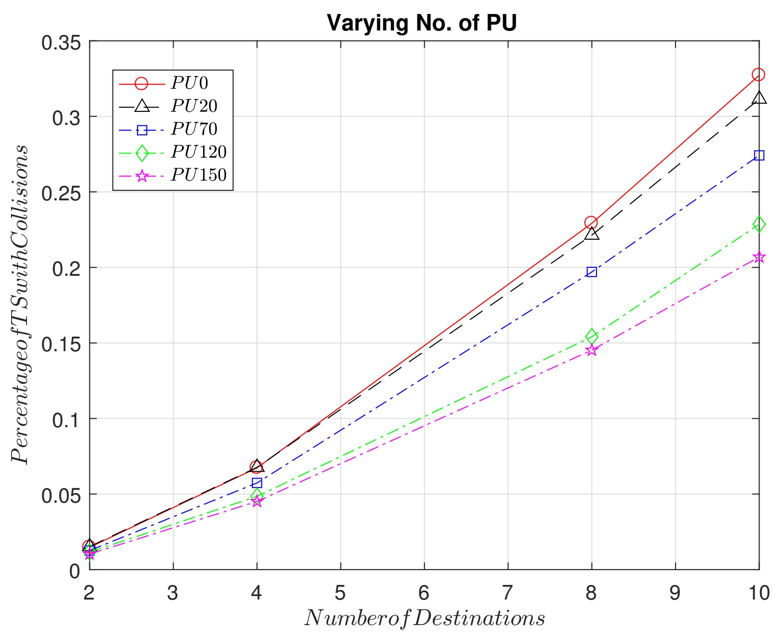


Figure 4.9: Percentage of time slots with collisions with the rendezvous of varying number of destinations

In Figure 4.10, we show the effect of varying number of present PUs in the system on the percentage of time slots with collisions. Considering the case of 4-destination rendezvous, the percentage of time slots with collisions varies from approximately 0.068% to 0.045% with the increase of number of PUs from zero - i.e. no PUs using the system channels at all - to 150 PU utilizing the system channels. Same effect goes for different cases when considering different number of destinations for multi-destination rendezvous as clear from Figure 4.11. The highest percentage of time slots with collisions is recorded when the PUs presence is the least in the system while the minimum recorded percentage takes place when the number of PUs in the system peaks to 150.

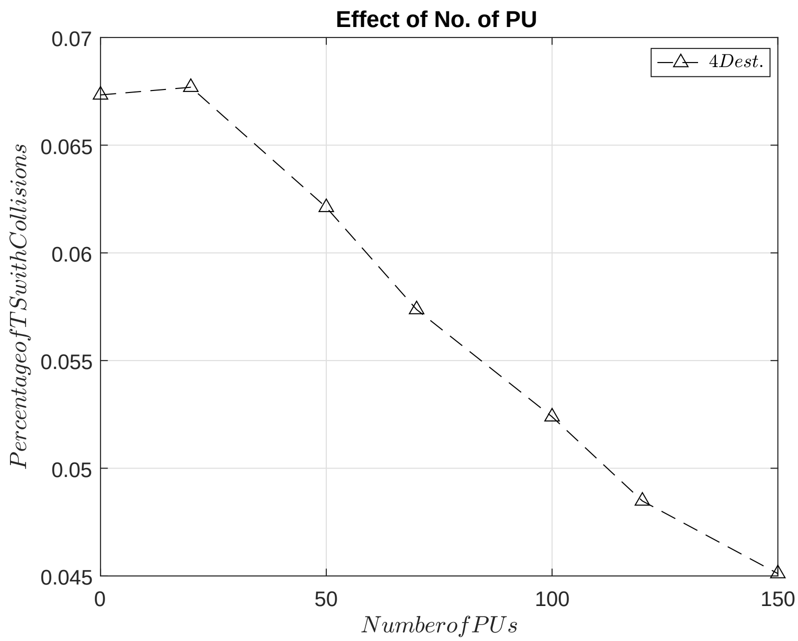


Figure 4.10: Percentage of time slots with collisions with the rendezvous of four destinations for varying number of PUs

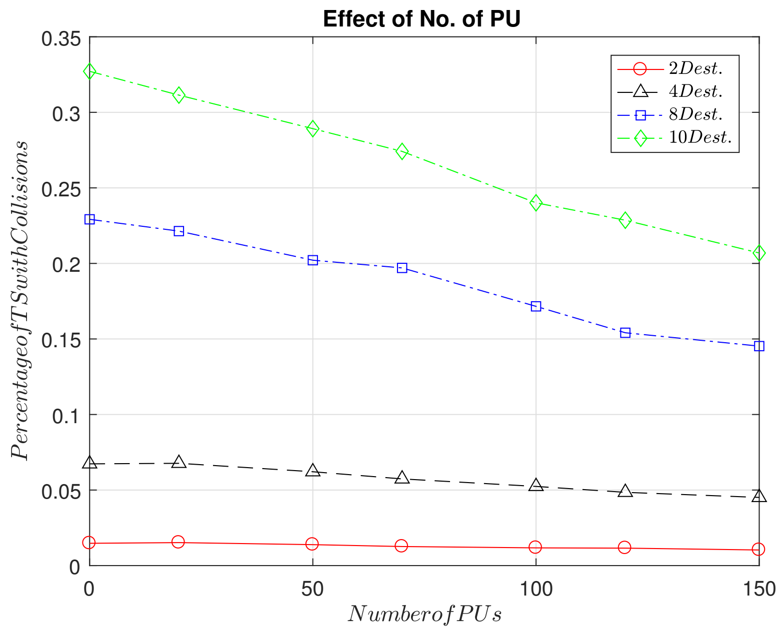


Figure 4.11: Percentage of time slots with collisions with the rendezvous of varying number of destinations for varying number of PUs

Such trend of decrease in the percentage of time slots with collisions with the increase in number of PUs may also seem extraordinary and against the expectations where usually the increase in number of PUs in a cognitive radio network is expected to affect the SUs negatively, but the variation in numbers is not of great significance. As explained earlier in section 4.1.1, with the increase in number of PUs, the available channel sets for SUs shrink leading to a better relative MTTR, compared to the traditional technique. Here the increase in number of PUs leaves less resources for SUs to use and therefore less availability for SUs to utilize the system channels. Considering the necessary conditions that need to be fulfilled for the collision to take place, as explained in section 3.5.1, it can be deduced that less resources for SU means tougher situation for SUs to be present on the same channels with their SUs which may mean a lower probability to achieve those necessary conditions to have the source SU and at least two of its intended destinations all together on the same channel at the same time and therefore slightly lower percentage of collisions.

A coincidence occurred throughout simulations that the randomly generated parameters *i, r and t* used with the Enhanced Jump Stay function that generates the hopping sequence *EJSH(i,r,P,M,t)*, are the same for two destinations (of the 8 or the 10 destinations for example), so these two keep colliding each time because their generated channel numbers will always be the same, thus continuously counting collisions and a failed rendezvous each time which can get the MTTR to go to infinity. This coincidence has occurred 390 times out of 2*242877=0.000802876, that is 0.08%, in another trial it occurred 578 times out of 2*242762=0.001190466,i.e. 0.12% per Full N destinations rendezvous trial on average, with max noticed is 2 times per Full N destinations rendezvous trial. This scenario and the method to overcome its destructive effect has been discussed in detail in section 3.5.3.

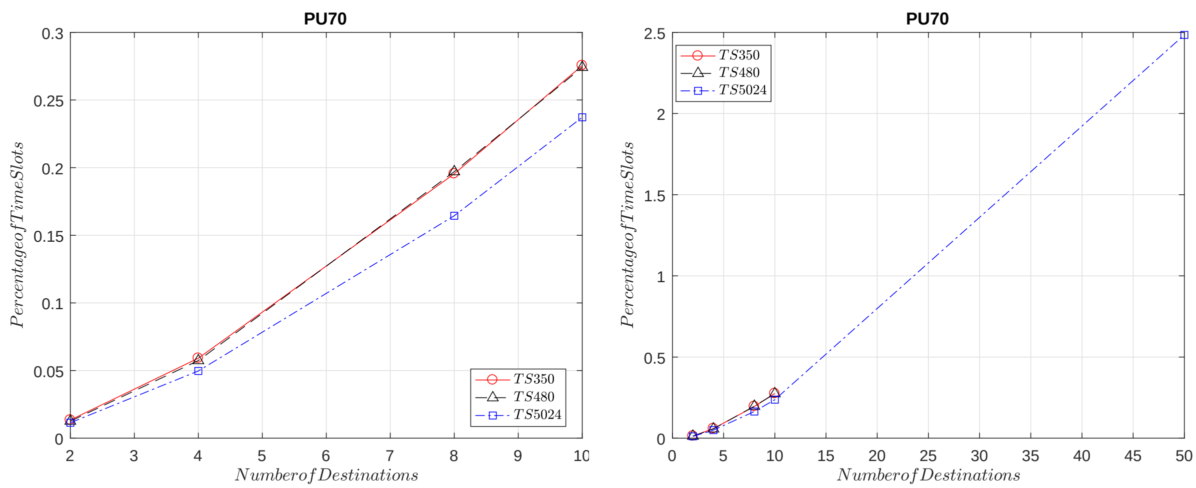
4.2.2 Effect of Time Slot Length

The experiments carried out in the current section were meant to analyze the effect of the time slot length on the percentage of time slots with collisions. Same simulation parameters apply from section 4.1.2 , and here as well, we vary the length of the time slot from 350 bits going up to the extreme length of 5024 bits and for such case of the longer time slot, we consider the case of 50 concurrent destinations as the extreme case in the simulation similar to section 4.1.2.

We started by simulating the behavior at the presence of an average number of PUs in the system which is 70. In Figure 4.12(a), the percentage of time slots with collisions is plotted against varying number of destinations for three different time slot lengths, 350 bits, 480 bits and the longer time slot of 5024 bits. It is obvious that the length of the time slot has very minor effect on the percentage of time slots with collisions, for example at 2 destinations, the percentage varies from 0.013% to 0.011% when the time slot length varies widely between 350 and 5024 bits (more than 10 times), similarly at the 4 destinations, it varies from 0.059% to 0.05%.

In Figure 4.12(b), we show the behavior at the extreme case of 50 concurrent destinations. As observed from the figure, the collisions' percentage continues to increase with the increase of number of concurrent destinations till it reaches 2.5% at the extreme case of 50 destinations, which is expected as the more the concurrent destinations considered, the more likely two of them would be on the same channel with the source SU at the same time satisfying the collision condition.

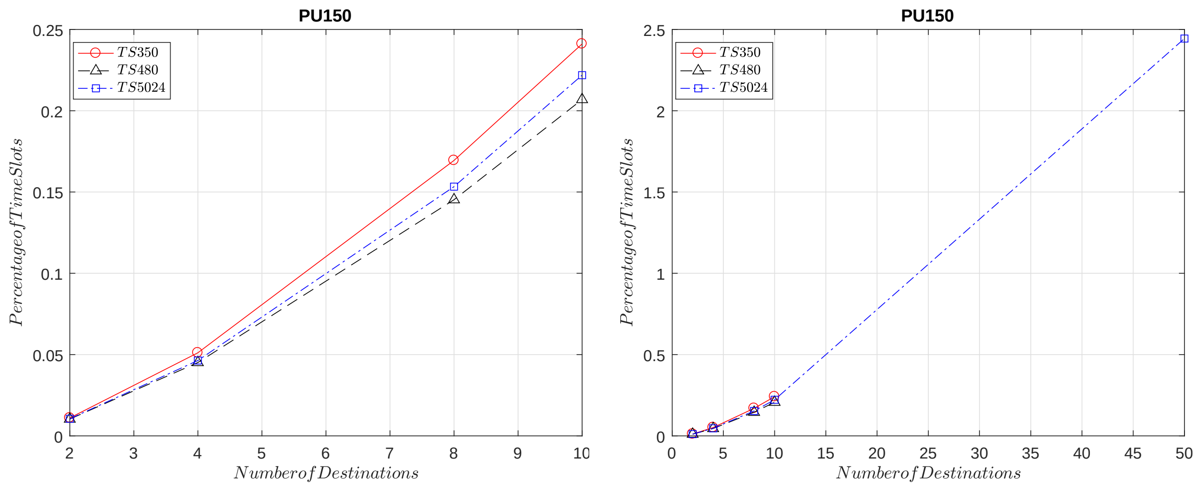
Hence we can deduce, as long as the system is time-slotted, the time slot length is not a major factor affecting the percentage of time slots with collisions during the rendezvous process.



(a) Up to 10 destinations (b) up to 50 destinations

Figure 4.12: Percentage of time slots with collisions with the rendezvous of varying number of destinations taken at 70 PUs

To further confirm that position, we also simulated the performance at the presence of a higher number of PUs in the system which is at 150 PUs.



(a) Up to 10 destinations

(b) up to 50 destinations

Figure 4.13: Percentage of time slots with collisions with the rendezvous of varying number of destinations taken at 150 PUs

In Figure 4.13(a), similar to the previous figure, the percentage of time slots with collisions is plotted against the varying number of destinations for the three different time slot lengths. At 2 destinations, the percentage varies from 0.011% to 0.0105% when the time slot length varies between 350 and 5024 bits , similarly at the 4 destinations, it varies from 0.0511% to 0.0464%. The collision percentage varies slightly for the rest of number of destinations values over the different time slot lengths, at 10 destinations, it varies from 0.24% to 0.22%.

In Figure 4.13(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, here as well, the collisions' percentage continues to increase with the increase in number of concurrent destinations till it reaches approximately 2.4% at the extreme case of 50 destinations.

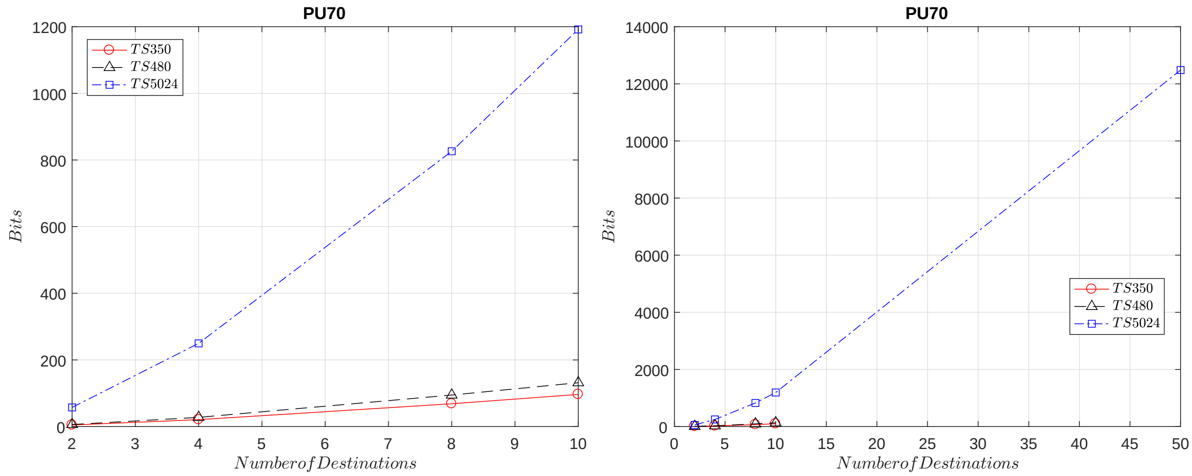
4.2.3 Bit Errors

In this section we focus on the bit errors rather than the collision behavior. Results presented in this section are based the outputs of the previous section and therefore on similar parameters to those used in section 4.1.2 also taken at the presence of 70

PUs in the system. Moreover, they are presented for the same lengths of the time slots 350, 480 and the extreme case of 5024 bits.

To give more sense to the outputs from the previous results, we further analyze by considering a 100 time slot interval and convert the collisions to the number of bits lost - bit errors - in 100 time slots. In Figure 4.14(a), the number of bits lost is plotted against varying number of destinations for three different time slot lengths, 350 bits, 480 bits and the longer time slot of 5024 bits. When comparing the bits lost for the same number of concurrent destinations at different time slots, it is obvious that as the length of the time slot increases, the number of bits lost due to collisions increases. For example at 2 destinations, the number of bits lost is 5 bits at the length of 350 bits, which increases to 7 bits when considering the longer time slot of 480 bits and abruptly jumps to 58 when considering the longer time slot of 5024 bits. Similarly at the 4 destinations, the number of lost bits is 21 bits at the length of 350 bits, which increases to 28 bits when considering the longer time slot of 480 bits and abruptly jumps to 250 bits when considering the longer time slot of 5024 bits.

In Figure 4.14(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, the lost bits continues to increase with the increase in number of concurrent destinations till it reaches the value of 1248 bits (156 Bytes) at the extreme case of 50 destinations.



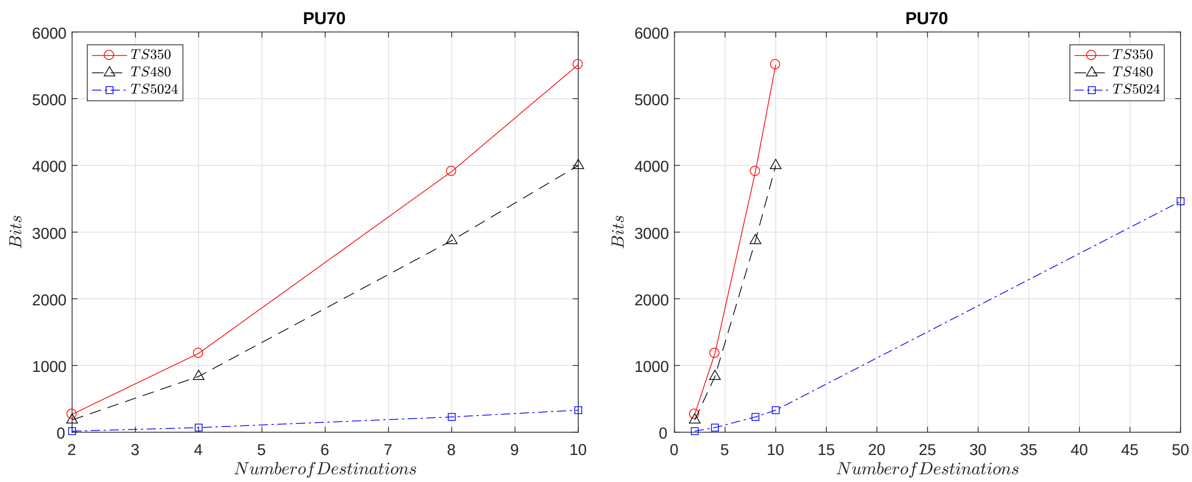
(a) Up to 10 destinations

(b) up to 50 destinations

Figure 4.14: The Bits lost in 100 time slots, with the rendezvous of varying number of destinations for different system time slot lengths

Moreover, we show the number of bits lost in a time interval of 1 second for the three time slot values. In Figure 4.15(a), the number of bits lost in 1 second of time is plotted against varying number of destinations for three different time slot lengths, 350 bits, 480 bits and the longer time slot of 5024 bits. When comparing the bits lost for the same number of concurrent destinations at different time slot lengths, it is obvious that the results here oppose those in Figure4.14, as the length of the time slot increases, the number of bits lost due to collisions in 1 second decreases. For example at 2 destinations, the number of bits lost is 269 bits at the length of 350 bits, which decreases to 184 bits when considering the longer time slot of 480 bits and abruptly drops to 16 when considering the longer time slot of 5024 bits. Similarly at the 4 destinations, the number of lost bits is 1183 bits at the time slot length of 350 bits, which decreases to 837 bits when considering the longer time slot of 480 bits and abruptly drops to 70 bits when considering the longer time slot of 5024 bits. This can be explained as follows, the time interval of 1 second has 5715 time slots of the 350 bits time slot compared to only 399 time slots of the longer 5024 bits time slot. In addition, considering the very low percentages of time slots with collisions for all the time slot values, we can understand why the 5024 time slot would give much lower lost bits value this time.

In Figure 4.15(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, the lost bits in 1 second continues to increase with the increase of number of concurrent destinations till it reaches a value of 3462 bits (433 Bytes) at the extreme case of 50 destinations.



(a) Up to 10 destinations

(b) up to 50 destinations

Figure 4.15: The Bits lost in 1 second, with the rendezvous of varying number of destinations for different system time slots

Here we have compared the number of bit losses to continue tackling the topic of the optimum system time slot length from another aspect which is the bit errors. Again, we observe the trade-off mentioned in section 3.3.3, using the traditional technique at any time slot length yields no such losses and of course irrelevant to any number of destinations the source SU might need to send to, but with our proposed protocol, those bit losses took place. Hence, it is obvious that utilizing our protocol, although it provides great enhancements as discussed throughout this chapter, it also comes with a downside which is collisions reflected in bit losses, and therefore, to further select a time slot length, consideration must be taken of the tolerable bit error rate by each application that will utilize the system.

4.3 Throughput

The throughput is defined as the number of completed transmissions per node per transmission time and is considered the most important key performance indicator (KPI) for any communication network. In the previous sections, we have showed the enhancements in the delay till rendezvous associated with the utilization of our protocol, as well as a downside of utilizing our protocol which is the collisions. In this section, we show another enhancement that our protocol provides with respect to throughput.

4.3.1 Effect of Number of Destinations and Number of PUs

Simulations presented in this section were carried out at the same simulation environment and using the same parameters as those of section 4.1.1, also the number of PUs vary within the simulation between zero and 150. To compare throughput between the case of utilizing the traditional technique and that of our protocol, the arrival rate should be forced to be maximized (saturated queues) such that, when utilizing our protocol (for example for a group of 4 destinations), the source SU should never remain idle waiting for the arrival of transmissions for the 4 destinations to be able to start treating them as group, but to always have many groups of 4s waiting in its queue instead of wasting time waiting for the queue to fill up and gather groups of 4s. In order to account for the aforementioned heavy load condition, the arrival rate was increased to 500 packets/sec for the SUs in order to always posses filled queues.

Figure 4.16 shows the percentage of increase in the throughput with the rendezvous of varying number of destinations. Our proposed protocol provides a significant enhancement over the traditional sequential technique. The enhancement increases with the increase in number of destinations undergoing trials to rendezvous simultaneously by the source SU. The percentage of increase varies from 15% when considering two destinations and goes up to 103% when considering an RTS that has 10 simultaneous destination addresses.

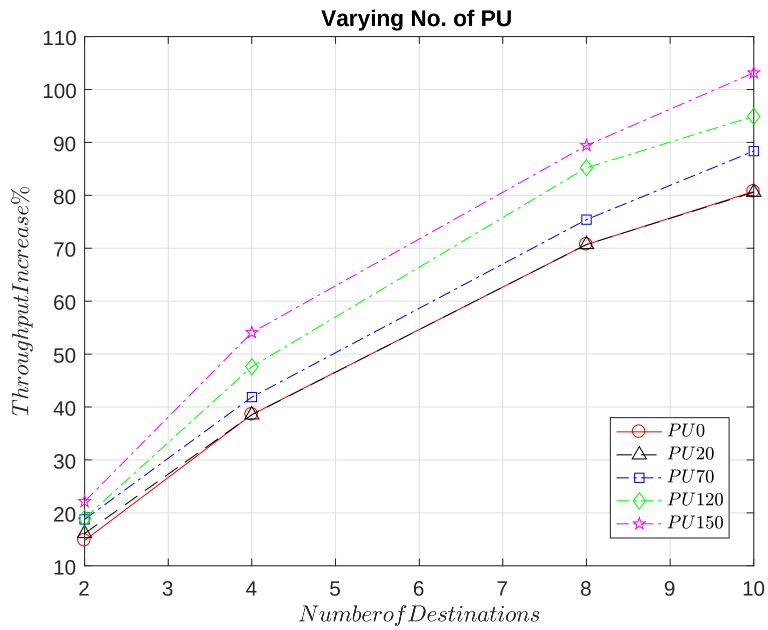


Figure 4.16: Percentage of Throughput increase with the rendezvous of varying number of destinations

In Figure 4.17, we show the effect of varying number of present PUs in the system on the percentage of increase of the throughput. Considering the case of 4-destination rendezvous, the increase varies from approximately 38% to 54% with the increase of number of PUs from zero - i.e. no PUs using the system channels at all - to 150 PU utilizing the system channels. Same effect goes for different cases when considering different number of destinations for multi-destination rendezvous as clear from Figure 4.18. The lowest percentage of throughput increase is recorded when the PUs are completely absent from the system while the maximum recorded percentage is that when the number of PUs in the system peaks to 150.

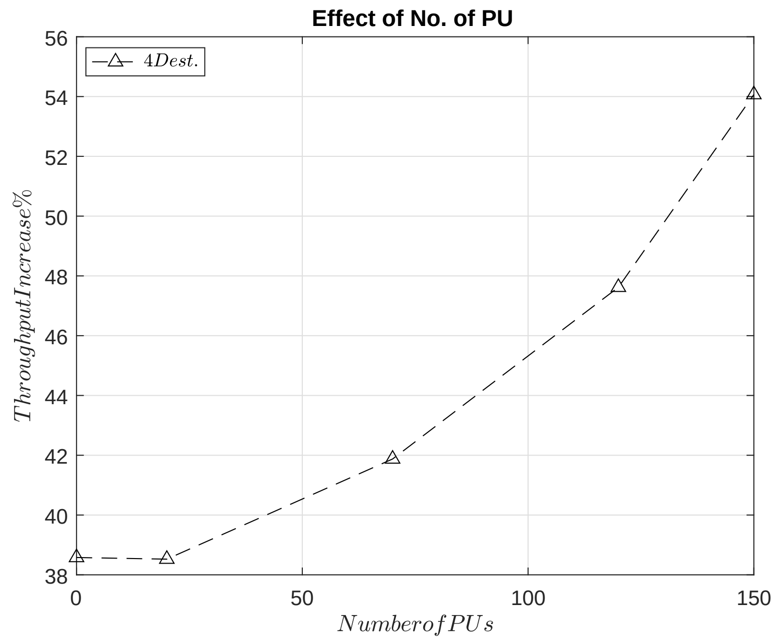


Figure 4.17: Percentage of Throughput increase with the rendezvous of four destinations for varying number of PUs

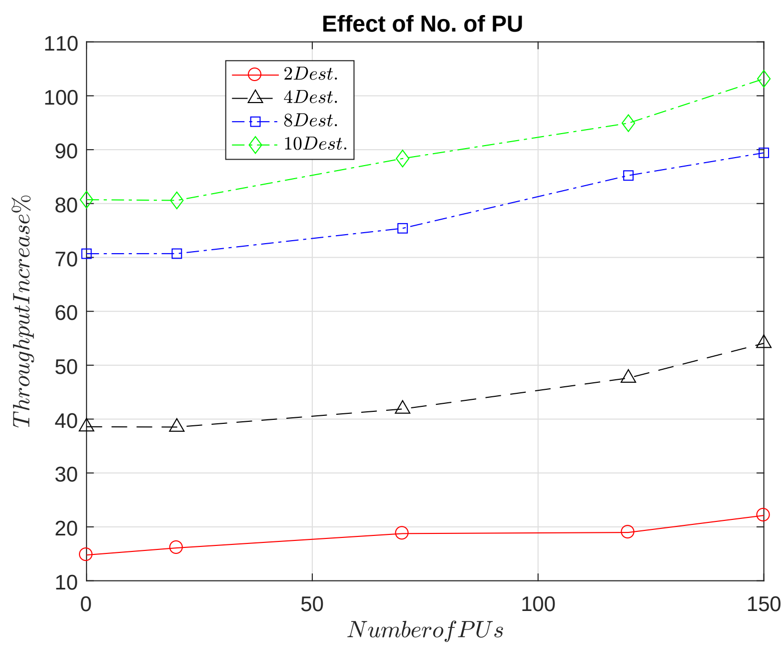


Figure 4.18: Percentage of Throughput increase with the rendezvous of varying number of destinations for varying number of PUs

Such trend of increase in throughput with the increase in number of PUs may seem extraordinary and against the expectations where the increase in number of PUs in a cognitive radio network is usually associated with negative effects on the SUs' communication.

As explained earlier in section 4.1.1, with the increase in number of PUs, the available channel sets for SUs shrink leading to a better relative MTTR, compared to the traditional technique. A better relative MTTR results in a better utilization of the system time as relatively less fraction of system time is spent in the rendezvous process and a relatively bigger time is spent transmitting SUs' data which in turn reflects in a better throughput in the network system, compared to the traditional technique. Here again our protocol exploits the problem of lack of resources and transforms it to an advantage to the SUs.

4.3.2 Effect of Time Slot Length

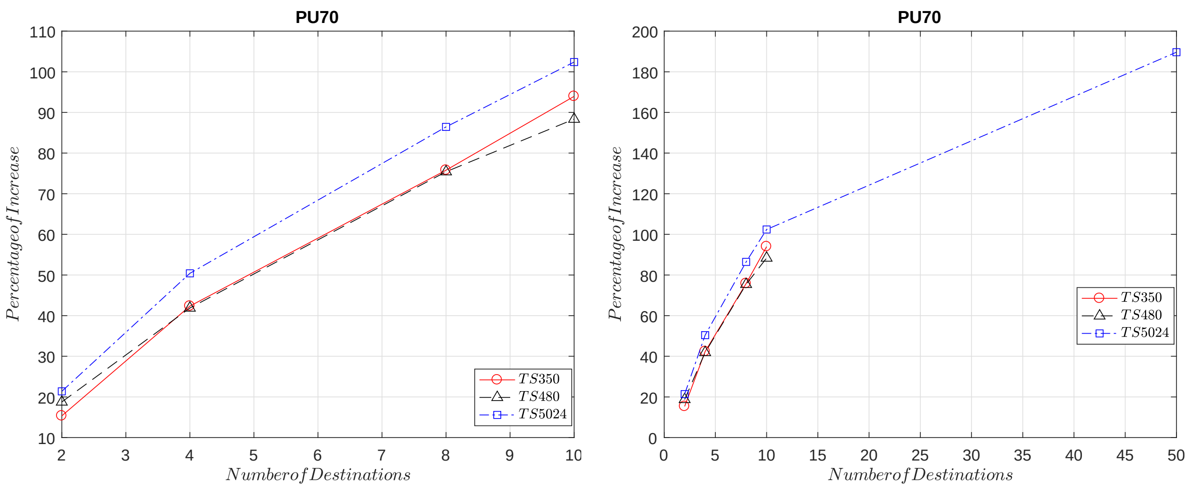
The experiments carried out in the current section were meant to analyze the effect of the time slot length on the percentage of throughput increase. Same simulation parameters are inhibited from the previous section, but we vary the length of the time slot from 350 bits going up to the extreme length of 5024 bits and for such case of longer time slot, we consider the case of 50 concurrent destinations as the extreme case in the simulation similar to section 4.1.2. For the longer time slot 5024, a smaller arrival rate was chosen to keep the network utilizations at similar value for fair comparison.

We started by simulating the performance at the presence of an average number of PUs in the system which is 70. In Figure 4.19(a), the percentage of throughput increase is plotted against varying number of destinations for three different time slot lengths, 350 bits, 480 bits and the longer time slot of 5024 bits. It is obvious that the length of the time slot has a minor effect on the enhancement percentage, for example at 2 destinations, the percentage varies from 15% to 21% when the time slot length varies widely between 350 and 5024 bits (more than 10 times), similarly at the

4 destinations, it varies from 42% to 50%.

In Figure 4.19(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, our proposed protocol shows great improvement over the traditional technique, the enhancement percentage continues to increase with the increase of number of concurrent destinations till it reaches more than 180% at the extreme case of 50 destinations.

Hence we can deduce, as long as the system is time-slotted, the trend of throughput increase compared to the traditional technique would be time slot length independent.

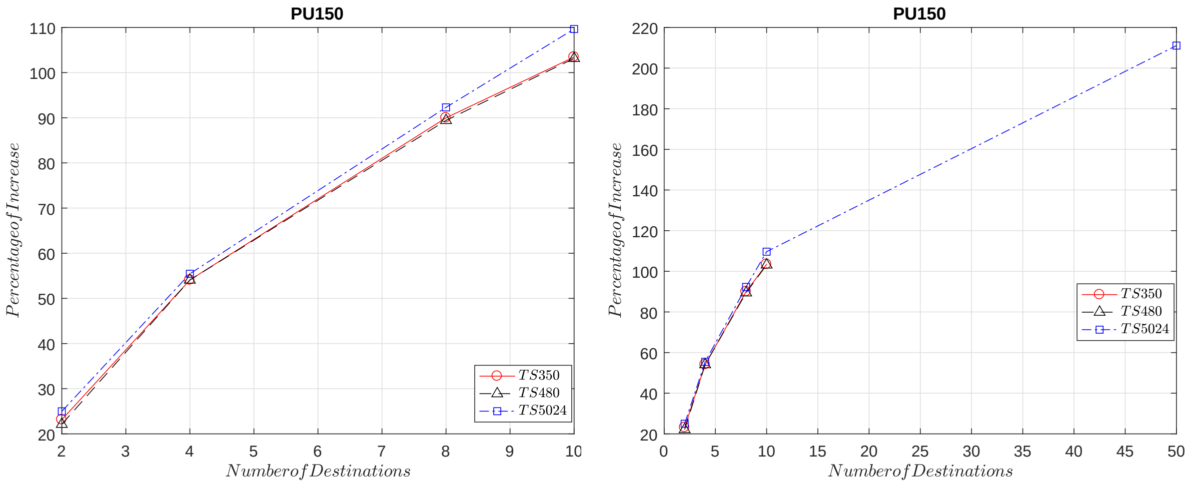


(a) Up to 10 destinations

(b) up to 50 destinations

Figure 4.19: Percentage of Throughput increase with the rendezvous of varying number of destinations taken at 70 PUs

To further confirm that position, we also simulated the performance at the presence of a higher number of PUs in the system which is at 150 PUs.



(a) Up to 10 destinations (b) up to 50 destinations

Figure 4.20: Percentage of Throughput increase with the rendezvous of varying number of destinations taken at 150 PUs

In Figure 4.20(a), similar to the previous figure, the percentage of throughput increase is plotted against varying number of destinations for the three different time slot lengths. At 2 destinations, the percentage varies from 23% to 25% when the time slot length varies between 350 and 5024 bits , similarly at the 4 destinations, it is varies between 54% and 55.5%. The percentage of increase varies slightly as well for the rest of varying number of destinations over the different time slot lengths, at 10 destinations, it varies from 103% to 110%

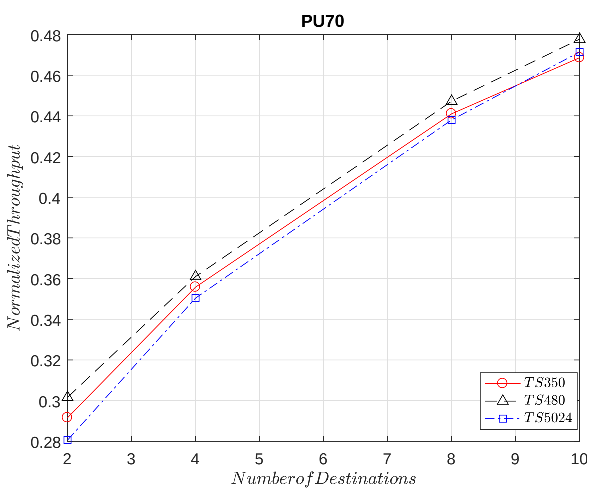
In Figure 4.20(b), we show the result of performance simulation for the extreme case of 50 concurrent destinations, our proposed protocol shows the vast improvement over the traditional technique here as well. The enhancement percentage continues to increase with the increase in number of concurrent destinations till it reaches more than 200% at the extreme case of 50 destinations.

4.3.3 Normalized Throughput

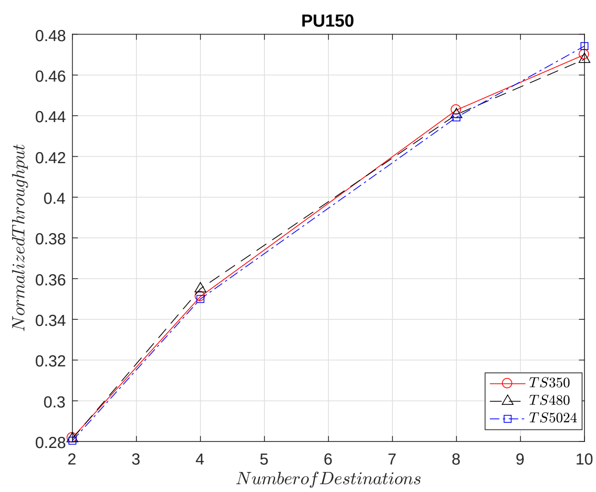
In this section we focus on the normalized throughput rather than the throughput increase. Results presented in this section were simulated using similar parameters to those used in the previous section and were performed for the same lengths of the time slots 350, 480 and the extreme case of 5024 bits. In addition, simulations were carried out at the presence of both an average number of PUs in the system which is 70 and a higher number of PUs in the system which is 150 PUs.

In Figure 4.21(a), the normalized throughput is plotted against varying number of destinations for three different time slot lengths, 350 bits, 480 bits and the longer time slot of 5024 bits at 70 PUs. Figure 4.21(b) is the normalized throughput as well but taken at 150 PUs.

As clear from both figures, the normalized throughput is totally independent of the length of the time slot and is almost not affected at all by the increase in number of PUs.



(a) At 70 PUs



(b) At 150 PUs

Figure 4.21: The Normalized Throughput with the rendezvous of varying number of destinations for different system time slots

CHAPTER 5: CONCLUSION AND FUTURE WORK

In this thesis, we have studied the problem of rendezvous of multiple destinations for the same source SU. We have presented the cognitive radio background and the research motivation. Moreover, we have studied the available literature and related work. Then, we proposed a multi-destination rendezvous protocol for blind rendezvous using single radio in cognitive radio networks. We presented the design considerations and the protocol behavior in different scenarios. After that we carried out a performance evaluation for the proposed protocol to show our contributions.

Among the open topics that will need further research in the future, is the length of the system time slot. The trade-off between the enhancement in performance achieved over a longer time slot and the delay that might occur in the rendezvous process to become even longer than the original MTTR of a sequential technique, the bit error rate, the fragmentation and more aspects are to be considered when selecting the length of the time slot. Therefore the selection of a certain time slot becomes a problem of optimization which should take into consideration all the aforementioned aspects.

Another topic which is also open to research is the design of the RTS and how to support more destination addresses. As per the limit imposed by the number of subtype bits in the RTS header format, the number of destination addresses that can be handled is limited to 34. Further research can suggest different techniques to overcome this limitation and allow the RTS to support more than 34 destination addresses.

REFERENCES

[1] M. Yglesias, "Us cell phone subscriber trends," November 2009. https://thinkprogress.org/us-cell-phone-subscriber-trends-4ee16d0a2a1e.

[2] N. Telecommunications and I. Administration, "United states frequency allocations: The radio spectrum chart," January 2016. https://www.ntia.doc.gov/page/2011/united-states-frequency-allocation-chart.

[3] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810–836, 2009.

[4] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[5] L. Gavrilovska, S. Member, D. Denkovski, S. Member, V. Rakovic, S. Member, and M. Angjelichinoski, "Medium Access Control Protocols in Cognitive Radio Networks : Overview and General," *IEEE Communications Surveys and Tutorials*, vol. 16, no. c, pp. 1–33, 2014.

[6] A. De Domenico, E. Calvanese Strinati, and M. G. Di Benedetto, "A survey on MAC strategies for cognitive radio networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 21–44, 2012.

[7] R. Rajbanshi, "Ofdm-based cognitive radio for dsa networks." *Dissertation Abstracts International* 68, no. 03, 2007.

[8] X. Liu and J. Xie, "A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks," *Proceedings - IEEE INFOCOM*, pp. 2085–2093, 2014.

[9] L. Xie, "Ecgr6188-8188 cognitive radio networks." *Lecture Notes, University of North Carolina Charlotte*, ch. 13, 2015.

[10] R. Gandhi, C. C. Wang, and Y. C. Hu, "Fast rendezvous for multiple clients for cognitive radios using coordinated channel hopping," *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks workshops*, vol. 1, pp. 434–442, 2012.

[11] C. F. Shih, T. Y. Wu, and W. Liao, "DH-MAC: A dynamic channel hopping MAC protocol for cognitive radio networks," *IEEE International Conference on Communications*, 2010.

[12] M. Altamimi, K. Naik, and X. Shen, "Parallel link rendezvous in ad hoc cognitive radio networks," *GLOBECOM - IEEE Global Telecommunications Conference*, 2010.

[13] C. Xin, M. Song, S. Member, and L. Ma, "Performance Analysis of a Control-Free Dynamic Spectrum Access Scheme," *IEEE Transactions on Wireless Communications*, vol. 10, no. 12, pp. 4316–4323, 2011.

[14] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," *Proceedings - 2010 IEEE/ACM International Conference on Green Computing and Communications, GreenCom 2010, 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing, CPSCom 2010*, pp. 755–760, 2010.

[15] C. Campolo, A. Molinaro, C. Casetti, C.-f. Chiasserini, and P. Torino, "An 802 . 11-based MAC Protocol for Reliable Multicast in Multihop Networks," *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pp. 0–4, 2009.

[16] J. Kuri and S. K. Kasera, "Reliable Multicast in Multi-Access Wireless LANs," *Computer Engineering*, pp. 1–11, 2001.

[17] S. Ray, S. K. Roy, D. Bhattacharyya, and T.-H. Kim, "A Theoretical Model to Support Multicast Involving Multiple Senders over DPMM in IEEE 802.11n," *2015 Fifth International Conference on Communication Systems and Network Technologies*, pp. 107–110, 2015.

[18] B. Yoon and H. S. Kim, "A Scheme Improving Performance of IEEE 802 . 11 Multicast Protocol," *INNOV 2012 : The First International Conference on Communications, Computation, Networks and Technologies*, no. c, pp. 15–18, 2012.

[19] A. Banchs, A. De La Oliva, L. Eznarriaga, D. R. Kowalski, and P. Serrano, "Performance analysis and algorithm selection for reliable multicast in IEEE 802.11aa wireless LAN," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3875–3891, 2014.

[20] C. T. Chang, C. Y. Chang, and S. W. Chang, "TMCP: Two-layer multicast communication protocol for Bluetooth radio networks," *Computer Networks*, vol. 52, no. 14, pp. 2764–2778, 2008.

[21] L. Farkas, B. Bakos, and P. Spanyi, "A Practical Approach to Multicasting in Bluetooth Piconets," *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 00, no. c, pp. 1360–1366, 2006.

[22] C. Cordeiro, S. Abhyankar, and D. Agrawal, "A dynamic slot assignment scheme for slave-to-slave and multicast-like communication in Bluetooth personal area networks," *GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, vol. 7, pp. 4127–4132, 2003.

[23] C.-h. Yang and S.-j. Lee, "Efficient Multicast Protocol for Supporting Mobile Nodes in Bluetooth Scatternets," *Journal of Information Technology and Applications*, vol. 3, no. 1, pp. 1–11, 2008.

[24] e. a. C. Chen T, Zhang H G, Maggio G M, "a Cluster-Based Cognitive Radio Network.," *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, vol. IEEE, 2007, pp. 168–178, 2007.

[25] H. A. Thomas and S. Xavier, "Trust Based Relay Selection Improves Security for Multiple Destinations in Cognitive Radio Networks," *International Journal of Engineering Science*, vol. 6, no. 5, pp. 4680–4683, 2016.

[26] H. Khalife, V. Conan, J. Leguay, and T. Spyropoulos, "Point to multipoint transport in multichannel wireless environments," *IEEE Wireless Communications and Networking Conference, WCNC*, pp. 1404–1409, 2013.

[27] H. Khalife, J. Seddar, V. Conan, and J. Leguay, "Validation of a point to multipoint cognitive radio transport protocol over GNU radio testbed," *IFIP Wireless Days*, 2013.

[28] A. E. Masri, N. Malouch, and H. Khalife, "A Routing Strategy for Cognitive Radio Networks Using Fuzzy Logic Decisions," *COCORA 2011, The First International Conference on Advances in Cognitive Radio*, no. c, pp. 13 – 18, 2011.

[29] C. Cordeiro and K. Challapali, "C-MAC : A Cognitive MAC Protocol for Multi-Channel Wireless Networks," *America*, pp. 147–157, 2007.

[30] T. Fujii, "Ad-hoc Cognitive Radio," *Communication*, vol. 1, no. i, pp. 589–592, 2005.

[31] H. Tran, T. Q. Duong, and H.-J. Zepernick, "Delay performance of cognitive radio networks for point-to-point and point-to-multipoint communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 9, 2012.

[32] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2002-Janua, no. c, pp. 3866–3875, 2002.

[33] G. Z. G. Zeng, B. W. B. Wang, Y. D. Y. Ding, L. X. L. Xiao, and M. Mutka, "Multicast Algorithms for Multi-Channel Wireless Mesh Networks," 2007.

[34] G. Zeng, B. Wang, Y. Ding, L. Xiao, and M. Mutka, "Efficient multicast algorithms for multichannel wireless mesh networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 1, pp. 86–99, 2010.

[35] Z. Yin, Z. Li, and M. Chen, "A Novel Channel Assignment Algorithm for Multicast in Multi-radio Wireless Mesh Networks," *2007 IEEE Symposium on Computers and Communications*, pp. 283–288, 2007.

[36] D. Shin, J. Kim, and Y.-b. Ko, "A Hybrid Topology based Multicast Routing for Cognitive Radio Ad Hoc Networks," *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, 2014.

[37] A. Raniwala, "Architecture and algorithms for an IEEE 802.1 1 -based multi-channel wireless mesh network," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, pp. 2223–2234, 2005.

[38] J. Qadir, A. Baig, A. Ali, and Q. Shafi, "Multicasting in cognitive radio networks: Algorithms, techniques and protocols," *Journal of Network and Computer Applications*, vol. 45, pp. 44–61, 2014.

[39] H. M. Almasaeid and A. E. Kamal, "Exploiting multichannel diversity for cooperative multicast in cognitive radio mesh networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 770–783, 2014.

[40] L. Yu, C. Liu, S. Hua, and M. Liu, "Cognitive radio assisted quality compensation for scalable video multicast in cellular networks," *Signal Processing: Image Communication*, vol. 29, no. 10, pp. 1092–1101, 2014.

[41] Q. Liu, D. Pang, G. Hu, X. Wang, and X. Zhou, "A neighbor cooperation framework for time-efficient asynchronous channel hopping rendezvous in cognitive radio networks," *2012 IEEE International Symposium on Dynamic Spectrum Access Networks, DYSPAN 2012*, pp. 529–539, 2012.

[42] H. M. Almasaeid, T. H. Jawadwala, and A. E. Kamal, "On-demand multicast routing in cognitive radio mesh networks," *GLOBECOM - IEEE Global Telecommunications Conference*, no. Ccc, 2010.

[43] W. Kim, S. Y. Oh, M. Gerla, and J.-S. Park, "COCAST: Multicast mobile ad hoc networks using cognitive radio," *MILCOM 2009 - 2009 IEEE Military Communications Conference*, pp. 1–7, 2009.

[44] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, p. 169, 2007.

[45] S. Bhadra and S. Shakkottai, "Looking at large networks: Coding vs. queueing," *Proceedings - IEEE INFOCOM*, vol. 00, no. c, 2006.

[46] Y. Qu, C. Dong, H. Dai, F. Wu, S. Tang, H. Wang, and C. Tian, "Network coding-based multicast in multi-hop CRNs under uncertain spectrum availability," *Proceedings - IEEE INFOCOM*, vol. 26, pp. 783–791, 2015.

[47] A. Asterjadhi, N. Baldo, and M. Zorzi, "A distributed network coded control channel for multihop cognitive radio networks," *IEEE Network*, vol. 23, no. 4, pp. 26–32, 2009.

[48] M. Z. Farooqi, S. M. Tabassum, M. H. Rehmani, and Y. Saleem, "A survey on network coding: From traditional wireless networks to emerging cognitive radio networks," *Journal of Network and Computer Applications*, vol. 46, pp. 166–181, 2014.

[49] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.

[50] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[51] D. Katabi, S. Katti, W. Hu, H. Rahul, and M. Medard, "On practical network coding for wireless environments," *International Zurich Seminar on Digital Communications*, vol. 2006, pp. 84–85, 2006.

[52] S. Katti, M. Muriel, and J. Crowcroft, "XORs in The Air : Practical Wireless Network Coding," *ACM SIGCOMM computer communication review*, pp. 243–254, 2006.

[53] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, "The Importance of Being Opportunistic : Practical Network Coding for Wireless Environments," *Proc 43rd Annual Allerton Conference*, vol. 1, pp. 431–436, 2005.

[54] D. S. Lun, M. M, and R. Koetter, "Network Coding for Efficient Wireless Unicast," *Science*, no. 1, pp. 74–77, 2006.

[55] A. Fanous, Y. E. Sagduyu, and A. Ephremides, "Reliable spectrum sensing and opportunistic access in network-coded communications," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 3, pp. 400–410, 2014.

[56] S. Wang, Y. E. Sagduyu, J. Zhang, and J. H. Li, "Spectrum shaping via network coding in cognitive radio networks," *Proceedings - IEEE INFOCOM*, pp. 396–400, 2011.

[57] Y. Li, H. Long, M. Peng, and W. Wang, "Spectrum sharing with analog network coding," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1703–1716, 2014.

[58] R. Gao and W. Hui, "A parallel rendezvous multi-channel MAC protocol for distributed cognitive radio networks," *2011 International Conference on Internet Technology and Applications, iTAP 2011 - Proceedings*, pp. 1–4, 2011.

[59] L. Yu, H. Liu, Y.-W. Leung, X. Chu, and Z. Lin, "Multiple Radios for Fast Rendezvous in Cognitive Radio Networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2014.

[60] Z. Lin, H. Liu, X. Chu, and Y. W. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1742–1745, 2013.

[61] X. Liu and J. Xie, "A slot-asynchronous MAC protocol design for blind rendezvous in cognitive radio networks," *2014 IEEE Global Communications Conference, GLOBECOM 2014*, pp. 4641–4646, 2014.

[62] C. Cormio and K. R. Chowdhury, "An adaptive multiple rendezvous control channel for cognitive radio wireless ad hoc networks," *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2010*, pp. 346–351, 2010.

[63] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," *2008 IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2008*, pp. 440–446, 2008.

[64] J. Shin, D. Yang, and C. Kim, "A channel rendezvous scheme for cognitive radio networks," *IEEE Communications Letters*, vol. 14, no. 10, pp. 954–956, 2010.

[65] K. Bian, J.-M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 25–36, ACM, 2009.

[66] C. Cormio and K. R. Chowdhury, "Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping," *Ad Hoc Networks*, vol. 8, no. 4, pp. 430–438, 2010.

[67] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 216–227, 2011.

[68] Y. Song, "Optimal secondary user packet size in mobile cognitive radio networks under fading channels," *Proceedings - IEEE INFOCOM*, vol. 26, pp. 163–171, 2015.