Frequency Hopping in Staggered Multi-Hopped Multi-Subnet Environment

By

Waqas Masud Khan

[2010-NUST-MS PhD-ComE-18]

Submitted to the Department of Computer Engineering In partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering

Advisor Col. Dr Shoab Ahmad Khan

College of Electrical & Mechanical Engineering National University of Sciences and Technology 2012

Declaration

I hereby declare that I have developed this thesis entirely on the basis of my personal efforts under the sincere guidance of my supervisor Dr Shoab Ahmed Khan. All the sources used in this thesis have been cited and the contents of this thesis have not been plagiarized. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

____

Signature

Waqas Masud Khan

(2010-NUST-MS PhD-ComE-18)

Acknowledgement

First of all, all Thanks to Almighty Allah the most merciful and beneficent, whose blessing and help made the completion of this thesis possible. I would also like to thank my parents for their unconditional support, care and patience throughout my life and especially during the course of my masters program. Words cannot express the gratitude that I owe them. My special thanks to my supervisor Col. Dr. Shoab Ahmad Khan who despite his hectic and busy schedule was always available for helping me and guiding me on my thesis. I feel honored to have worked under the supervision of such a world class engineer and scientist. Due to his vast knowledge and guidance I was able to successfully complete this thesis. I would also like to thank my senior and lecturer Sajid Gul Khawaja for both his moral and technical support and help throughout the course of this thesis.

Abstract

Multi-Hopping is becoming increasingly important method for routing data among nodes in wireless Ad-Hoc networks due to the fact that Ad-Hoc networks are usually setup in real time without any specific configuration. The multi-hopping techniques used in wireless Ad-Hoc networks along with frequency hopping have several advantages in today's world that makes it ideal for situations in which security is of utmost importance. This is in part due to the fact that frequency hopping has inherent immunity from jamming and eavesdropping which is increased in a staggered multi-hop multi-subnet environment. Due to the increasing importance of multi-hopping in wireless Ad-Hoc networks new algorithms are beginning proposed regularly on all levels of the ISO layer stack.

In this MS Thesis we have developed and implemented a new technique for frequency hopping in staggered Multi-Hop Multi-Subnet environment. The method implements the synchronization of frequency hopping and time division multiple access in a staggered multi-hop multi-subnet environment for an Ad-Hoc network. The technique implemented in this paper works on the ISO Datalink layer (layer 2) and is used initially for synchronizing source, destination radios and two hops between them for different groups of radios in different subnets out of n radios, during this time all of the radios are communicating in different time slots on the same frequency hopping pattern. Then it moves on to synchronize the transfer of data between the groups of radios in different subnets where the radios in the different subnets communicate with their own staggered frequency hopping pattern for multiple access which further strengthens the security of such networks. In our simulations we have successfully synchronized frequency hopping in staggered multi-hopped multi-subnet environment among n radios.

Table of Contents

List of Figures

List of Equations

Introduction

Chapter 1

1.1 Background

This Thesis is the research and implementation of a new technique for frequency hopping in a staggered multi-hop multi-subnet environment. Research and development in wireless communications has seen an explosive growth in the past 15 years in the fields of fixed wireless networks with infrastructure and a backbone. But now much work is being done in wireless networks without infrastructure for example wireless sensor networks and mobile multi-hop Ad-hoc networks.

Frequency Hopping a spread spectrum technique is one of the most important means for digital wireless communications and is achieved by randomly switching the carrier frequency during transmission among a predefined range of frequencies. The pseudorandom sequence by which the carrier frequency randomly switches from one frequency to another is known by both the transmitter and receiver. It is also utilized as a multiple access methods used in many digital communication systems and standards worldwide.

Spread-spectrum transmission techniques have many advantages in comparison to a fixed-frequency transmission:

1. The Frequency Hopping Spread Spectrum technique has a strong resistance towards narrowband interference. When a spread signal is recollected it spreads out the interfering signal which in effect causes it to recede into the background.
2. The Frequency Hopping Spread Spectrum transmissions are difficult to intercept. To a narrowband receiver a frequency hopping transmission appears as an increase in background noise. Also a frequency hopping spread spectrum transmission could only be detected by an eavesdropper if the pseudorandom sequence is known.
3. The same frequency band can be shared by frequency hopping spread spectrum transmissions and spread spectrum transmissions in general alongside many other traditional/conventional transmissions with minimal interference. This in effect utilizes the bandwidth more efficiently.

Due to its immunity towards jamming, interception, detection, multipath inference and multiple access capability frequency hopping has become a very popular method for digital communication and spread spectrum. For these following reasons frequency hopping is used in a number of applications for communication systems from highly secure radios for military communications to Bluetooth communication among cell phones. Link 16 which is a military tactical data exchange network also uses frequency hopping and the frequency hops 77,000 times per second in Link 16.

Wireless networks in use today are of two broad categories one is wireless networks with fixed infrastructure and the other is wireless networks without infrastructure. An example of a fixed wireless network is the GSM cellular network which has Base Transceiver Stations, Base Switching Center and at the backend has Mobile Switching Center all of which are fixed. All of the communication is eventually routed through the backbone of such a network. Whereas, in a wireless network without any infrastructure the wireless devices can communicate directly with each other if they are in range of each other. Otherwise data could be routed to the destination through multi-hopping. Multi-hop Ad-Hoc networks are wireless networks without any infrastructure. A lot of research has been carried out in the field of multi-hop communications for Ad-Hoc networks. Ad-Hoc networks are used in everything from wireless sensor networks in Industrial Installations to software defined radios working as node used in Ad-Hoc networks in disaster situations. In order to transfer data and information from source to destination such systems use multi-hopping. In multi-hopping the data hops from one node to another until it reaches its destination. As data is received at one node it is reproduced and then retransmitted to another node. Due to the fact that these nodes do not have to be in a specific layout or in any specific format they are usually deployed with only a small amount of prior planning. Also routes can be discovered through multi-hopping by different routing protocols which in effect make multi-hopping a very popular method of communication for Ad-Hoc networks. In such multi-hop Ad-Hoc networks nodes can enter the network and leave it randomly. Due to this property of multi-hop Ad-Hoc network the topology and layout of multi-hop Ad-Hoc networks is constantly changing. This creates many challenges for synchronization of both time divisions multiple access and frequency hopping in staggered multi-hops and among multi-subnets in such an environment and network. Thus all of the nodes constantly transmit frame for initializing communications after a certain time interval. Also issues are created because of the limited capabilities of the nodes in terms of processing power, battery life etc. For example in a wireless sensor network a node consists of a sensor, a small processor, memory and a transceiver for communications. In order to monitor a mountain for landslides or a volcano for an eruption many nodes have to be deployed. The data is usually feedback to the base station through multi-hopping. Another reason for using multi-hopping is because it saves transmission power in such a scenario. Instead of transmitting data and information over a large distance it only has to send it over a small distance. Multi-hopping in multi-subnets also is preferred in situation in which security is of utmost importance as data is transferred only among the nodes that are a part of the subnet. This type of multi-hopping is used in radios and communication devices aboard military vehicles.

1.2 Foundation of Thesis

This Thesis is the study and research of a proposed technique implemented for frequency hopping in a staggered multi-hop multi-subnet environment. It is implemented on the ISO Datalink Layer (Layer 2). In the proposed technique a new method for frequency hopping is implemented also the synchronization of frequency hopping and time slots among multi-hops and multi-subnets for n radios. Analysis has also been conducted in the frequency domain to shown the hopping of the frequency.

1.3 Organization of work

This thesis is organized into five chapters. A brief description is as follows,

Chapter 1: Provides an introduction, background and layout of the thesis.

Chapter 2: Provides a Literature Survey which includes related work and summary.

Chapter 3: Provides a detailed description of the proposed technique.

Chapter 4: Presents the evaluation and results.

Chapter 5: Presents Conclusion and future work.

Literature Survey and Review

Chapter 2

2.1 Related Work

Frequency hopping (FH) and direct sequence (DS) spread spectrum techniques are some of the most important means for digital wireless communications; this is due to their immunity towards jamming, interception, detection, multipath inference and multiple access capability. Several different techniques have been previously implemented to perform frequency hopping and to make the overall system more efficient by using frequency hopping. In [3] a frequency hopping technique in which admission into the frequency hopping sequence is controlled by a admission control technique in which C/I (Carrier to Interference) of a cell in a GSM system is maintained in order to preserve speech quality. This is performed in [3] by using the traffic load admission algorithm in two traffic environments this is a uniform environments and hotspots environment. In [4] a frequency hopping technique that avoids interference from different systems is used. A wireless communication system which is operating in an independent band usually avoids interferences from different systems because it is specified to operate in that one specific band. The ISM (Industrial, Scientific and Medical) band is duty free and is universally available for use by many wireless communication systems that's why a wireless communication system that is operating in it can easily suffer from many different sources of interference. As shown by [4] in order to avoid such circumstances in communication devices like Bluetooth that operate on the ISM band and use FHSS (Frequency Hopping Spread Spectrum) a cognitive frequency hopping technique can be used . Several advantages of using Frequency Hopping over other techniques in Personal Communication Systems and design are given in [5]. In [1] a simple synchronization technique for synchronous-access FH transceivers is proposed. Where as in [6] a FH-code phase technique for synchronization in Multi-Hop FH/DSSS ADHOC Network is used. In [6] specified DS-code are used to spread the synchronization data, in such a system the code space contains the control channel. Messages are sent for synchronization and they are broadcasted periodically to allow single nodes or sub networks with different FH-code phases to synchronize themselves as a single network. Frequency Hopping systems have a heavy burden of synchronization and this to some extent can limit the amount of information exchange in [7] an innovative technique for Message Driven Frequency Hopping (MDFH) system is proposed. The spectral efficiently of a FH system can be increased by embedding a portion of the information into the hopping frequency selection process. This is done in [7] by using a Message Driven Frequency Hopping system rather than pseudo random sequence driven system as in conventional systems. Studies and research has also been conducted in order to show the benefits of Dynamic frequency hopping in cellular systems like GSM over random frequency hopping in [8]. Such studies in [8] show that if performance of the system is defined as 0.02 word error rate and this is for 90% of users in the system then dynamic frequency hopping gives 100% increase in capacity over fixed channel allocation also a 50% increase in capacity over random frequency hopping. These studies and simulations that were conducted in [8] analyze the performance of DFH (Dynamic Frequency Hopping) for a cellular system. In [8] the analysis is performed with a cellular system with certain parameters that are stated in [8] such frequency reuse and antenna sectorization. In [9] an adaptive frequency hopping method is proposed, that switches to a new channel only when the packet drop rate of that given channel falls below a certain threshold. The results that were obtained in [9] by simulations show that adaptive frequency hopping improve the network reliability by 10% to 20% than slotted frequency hopping.

Multi-Hop Ad-Hoc networks consists of nodes that are capable of transmitting, routing and receiving data in almost any different configuration. Multi-Hop Ad-Hoc networks are being increasingly used in everything from software defined radios used in disasters situations and defense operations by armed forces to personal area networks among laptops and wireless sensors. Several studies and research has been performed on Multi-Hop networks in [10] the performance of the Multi-Hop Ad-Hoc network with increase in scalability and capacity was tested using a Testbed of Portable Multi-Hop Ad-Hoc nodes using the Optimized Link State Routing Protocol (OLSR) configured in two layouts one was as a straight chain and the other was in a mesh topology. As the number of hops was increased the delay was increasing also as the number of hops increased the packet delivery rate deteriorates rapidly for more than 3 hops and for higher dissemination rates it deteriorated rapidly after one hop. In [10] the reference states that Multi-Hop Ad-Hoc networks should be limited to 3 hops in practical scenarios for performance. [11] Analyzes the effects different hopping strategies in multi-hop wireless networks the three strategies that are used in relay selection are most distant neighbor, closest neighbor and random neighbor. In the analysis [11] they were using the Multi-Hop aggregate information efficiency (MIE_A) as a metric to get several properties of the network. In the analysis [11] found that keeping the nodes at constant minimum power W_{min} gave the maximum MIE_A for all of the three strategies. Also the closest neighbor gave a much better MIE_A then the other two strategies used in relay selection in a multi-hop network. In [12] a wireless network Ad-Hoc routing protocol is designed which works alongside a centralized cellular network as in an emergency situation many paths of the centralized network may be down the network is capable of routing data from the terminal to the base station the terminal switches to Ad-Hoc mode when it is not able to access the base station directly. According to [12] it took only 3 Hops for 90% of the terminals to reach the BS and 20% of the terminals were able to reach it directly in various simulations.

Much attention is being given to conserving energy and power in a multi-hop network Ad-Hoc as almost all wireless nodes in a multi-hop network Ad-Hoc are battery powered. In [13] energy used by a multi-hop network is compared with the energy used in a single-hop network. In the research of reference [13] it is proved that for that specific model that a multi-hop network consisting of more than two hops is not feasible in energy limited wireless sensor network. [13] Also takes in account both the circuit energy consumed and transmission energy consumed. Where as in [14] a routing protocol is proposed which used round robin technique to evenly distribute power on all sub-paths. In [15] and [16] allocation scheme and architecture for multi-subnet are described to improve their specific performance.

2.2 Literature Review

2.2.1 Frequency Hopping

It is the process of changing the carrier frequency that the transmitter is transmitting from one frequency to another frequency in a pre-defined or stored range of frequencies after a specific time interval. Frequency hopping is a Spread Spectrum technique that although has a large bandwidth it only uses a small portion of it during a specific time. Frequency hopping was initially used by military communication systems as it has a natural immunity towards jamming. But now is used widely in many digital communication systems and standard worldwide. Digital communication systems that use frequency hopping are resistant to narrowband interference. Figure 2.1, below shows a visual description of frequency hopping from one carrier frequency to another as time increases,

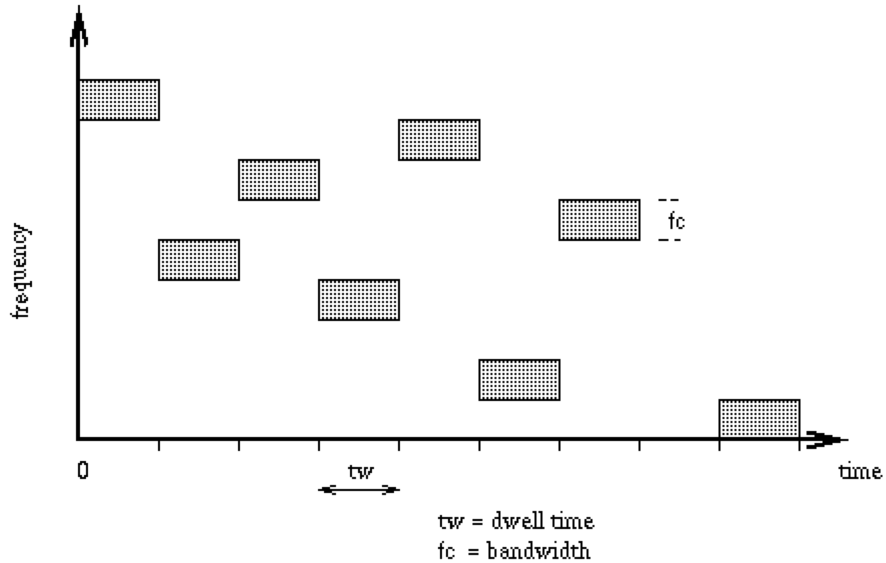


Figure 2.1 Frequency hopping visual diagram [18]

2.2.2 Time Division Multiple Access

Time Division Multiple Access is a method used by networks that share the same medium for transmission by allocating time slots to different users. By this technique several nodes of a network can share the same Rf frequency channel for transmission by transmitting only in their own allocated timeslots. Time division multiple access is similar to time division multiplexing but instead of having only one transmitter and one receiver there are several transmitters and receivers. Each transmitter only transmits data in its specified time slot which prevents interference among different radios in a communication system. These time slots are usually enclosed into frames that are repeated in a data stream. Figure 2.2, below shows a visual description of time slots enclosed in frames and frames in a data stream,

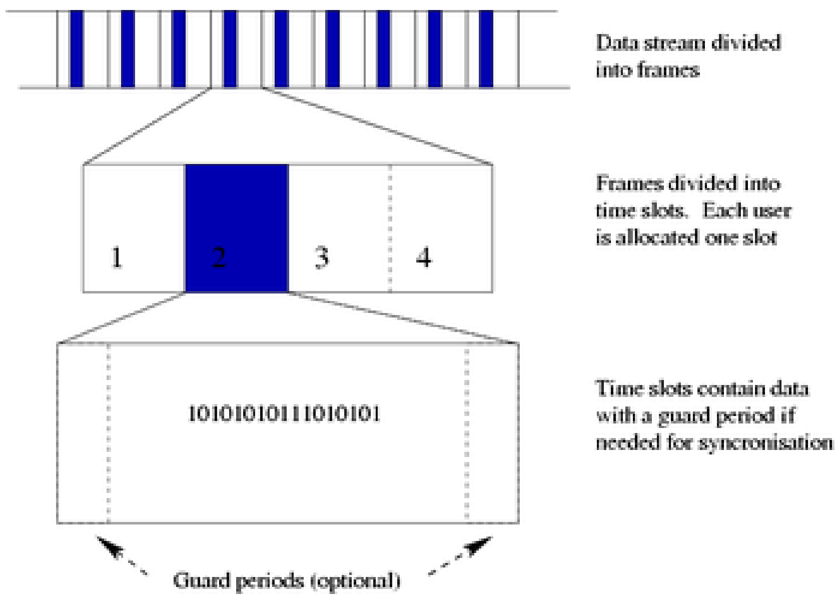


Figure 2.2 Visual diagram of Time slots in frames and data stream [19]

Time division multiple access is used alongside frequency division multiple access in wireless telecommunication systems worldwide. Time division multiple access is used in several digital communication standards worldwide it is used in almost all 2G systems except IS-95 it is used in the following 2G systems,

- Global Systems Mobile (GSM)
- D-AMPS
- Personal Handy-phone System (PHS)
- iDEN
- Personal Digital Cellular (PDC)

Although 3G systems mostly use CDMA, Time division multiple access is used in TD-SCDMA in 3G standard. TD-SCDMA is the Chinese national standard for 3G and it was developed by the China Academy of Telecommunications Technology (CATT) with cooperation from Siemens.

Characteristics of TDMA:

- Carrier frequency is shared by all users
- Handoff are simpler because of non-continuous transmission
- Slots are allocated on request in dynamic TDMA
- Power control is lesser than in CDMA
- Slot allocation complexity
- TDMA requires more synchronization than CDMA

In several wireless standards like Bluetooth, HIPERLAN/2 (broadband radio access network), WiMax are using dynamic time division multiple access which is a scheduling algorithm that based on the amount of traffic in a data stream it reserves a variable number of time slots for each frame based on a variable data stream.

Advantages of TDMA:

There are several advantages of TDMA systems due to its efficiency and characteristics. Some of the advantages are given below,

- TDMA is well suited for basic communication services such as voice, short message service and also bandwidth intensive services such as video conferencing.
- TDMA is immune from interference that is caused by several users using the same carrier frequency at the same time. This type of interference is however present in systems using spread spectrum techniques.
- TDMA saves more power since mobiles are only transmitting in their specified time slots.
- Handoffs in cellular systems are easier to perform because the transmission is not continuous.
- TDMA/GSM can handle 3 times as many calls as compared to FDMA

Disadvantages of TDMA:

There are also many disadvantages of TDMA some are the following,

- TDMA requires much more synchronization as compared to CDMA.
- If there are no free time slots in a cell in which a mobile phone is moving for a handoff then the call can be disconnected.
- If there are no free timeslots in a cell in which a mobile phone wants to make a call than it will be not be connected.
- Systems that use TDMA suffer from Multipath interference. A solution to such a problem is to have the signal received in a specific time frame but even at a thousandth of a second it still causes problems.

2.2.3 Hash Function

A Hash function is an algorithm or subroutine that maps a large dataset called a key of a variable or fixed length to a smaller dataset of fixed length. The output of a hash function is called a hash sums, hash values, or hash codes. Hash functions are not reversible. If H(x) is a hash function and x is a variable length input then the hash value would be y this is given in the equation 2.1 below,

$$y = H(x) \qquad \text{(Equation: 2.1)}$$

Hash functions are used by UNIX to store the location of a process, hash functions can also be used to generating indexes of an array, RAM or a lookup table. Hash functions are also heavily used in data comparison and detection tasks such as detecting duplicates records in a large database. There are hash functions that map different keys (inputs) to the same hash value thus causing collisions. Developing a good hash function is still a topic of active research. The following are some of the properties of what constitutes a good hash function,

- Continuity

  Hashing functions should be continuous that is if given two keys that have almost equal value the hashed value should be similar or close to each other. This is especially necessary in hash functions that are used to compare data for similar type of data. But continuity is only desired in some applications.

- Uniformity

  Hash functions should output hash values over a uniform range. That is a good hash function should have almost equally distributed hashed values and the probability of outputs should be roughly the same. This is because collisions among hashed values increase the costs of a hash function. The chi-squared test is used to evaluate the uniformity of a hash function.

- Determinism

  Hash functions should be deterministic that is for a given input they should always generate the same hash value. This is does not include hash functions that depend on external variables such as time.

- Low Cost

  The cost of computing a hash function should be less than the alternative of not using a hash function otherwise it would not be feasible to use a hash function. As in the case of searching a large database for similar records without a hash function should cost more in terms of computational complexity and processing power for it to be feasible to use a hash function. Also the hash function should produce a lesser amount of collisions and should be complex enough to handle the task.

- Data Normalization

Hash functions should be able to give the same hash value (output) given the same input. However in some applications the input data could be in different formats for example some letters could be in uppercase and others could be in lowercase. Hash function should have a built in data equivalence criterion so any two input that are in a different format should be normalized so that no ambiguity would remain.

- Variable Range

There are specific applications in which a different range for the output of a hash function is required in different runs of the application in such cases the hash function should take two inputs one input data and the other should be the input range. A common solution to very large hash value outputs for example from 0 to 2^{32} is to divide the output by n.

- Dynamic Hash Function

Dynamic hash functions are used in order to adjust the length of a hash table if a hash function needs to expand or shrink the length of the hash table during a run of an application.

Algorithms for Hash Functions:

The algorithms used for different types of hash functions depends upon the application in which it is used and the probability distribution and nature of the input data.

Trivial Hash function:

These are hash functions in which the input is small enough to have a different interpretation of it such as binary form in order to represent a hashed value of it. A trivial hash function is a perfect hash function as it maps every input value to a unique ash value. For example a 8 bit ASCII character represents text in computer and communication systems. ASCII represents all of the alphabets both upper case and lower case along with several other control characters in 2^8=256 bits. Whereas, in the case of Unicode characters it would take 17x2^{16} =1114112 bits to encode the same characters and letters. Country code that consists of two letters can be mapped by the same binary technique for example “uk”, “au” we get 26^2=676 bits.

Perfect Hash:

A Hash function that perfectly maps a valid input to a unique has value is considered a perfect hash function. In such hash functions no additional searching is required in order to locate an entry.

Minimal Perfect Hash Functions:

Minimal perfect hash functions are those hash functions that map an input onto a set of consecutive integers. If there are a total of n keys then the range of a minimal hash function is usually from 0 to n-1. Minimal hash functions are much lesser in number as compared to other perfect hash functions that are non minimal perfect hash functions.

Hash function for uniformly distributed data:

In applications which deal with bounded inputs for example car license plates, telephone numbers and invoice numbers. In such applications the probability of an input is equally likely to occur and such hash functions map an equal number of inputs to a hash value. For example if we have an input n in the range of 0 to N-1 and the output of the hash function is in the range of 0 to Z-1 the for an output H, a suitable hash function could be H=nmodZ.

Hash Functions with other distribution as input data:

In application which the input is not equally likely more complex hash functions are used. Also such hash functions should be just as sensitive to the last entries of an input as they are to the initial entries of an input.

Hash function for variable length data as input:

When a hash function takes a variable length data as an input for one input after another in a single run of an application then the resulting distributions can be every uneven. In such a case the hash function should be affected by ever individual character that it takes as an input in a string. Each individual character affects the hash value in a different way. This is every common in applications that search text documents such as emails.

Rolling Hash Functions:

Rolling hash function is generally used in sub-string search applications. For example if there is a string of N characters named S1, then if k is a fixed length integer such that k=N.

Now the hash function should search every sub-string s of fixed length in order to determine the hash value. K.N operations would be required in such a case.

Cryptographic Hash functions:

There are several cryptographic hash functions like SHA-1 that guarantee more uniform hash values than many other techniques such as fingerprinting and checksum. But the problem with such cryptographic hash function is that their cost is too much for ordinary applications. Cryptographic hash functions are used in applications such as denial of services attacks prevention.

2.2.4 Modulation

In Telecommunications modulation is the process of adding information into the carrier signal by varying one or more of the properties of the carrier signal. Information can also be transmitted through voice, hand signs and smoke signals but due to the lack of range of these methods we use telephone wires or transfer information through radio waves. The process of translating binary data so that it could be transferred over a wireless medium is called modulation. The Basic techniques for modulation are the following three,

- Amplitude Shift Key
- Frequency Shift Key
- Phase Shift Key

In all of these three methods one parameter of a sinusoidal is changed in order to represent a binary digit. These parameters are amplitude, frequency and phase and in amplitude modulation the amplitude is the parameter. In frequency modulation frequency is the parameter whereas in phase modulation the phase is the parameter. Voice signal is initially converted into an electrical signal than that sinusoidal is sampled and quantized. Once we have a sampled and quantized signal we can translate this binary signal of ones and zeros into a modulated signal by using any of the above three basic techniques. This modulated signal is than transmitted to the other side and at the receiving end the signal is demodulated and a near copy of your voice is reproduced. The carrier signal is the sin wave and the medium can be radio waves, wire, cable etc. The signal is usually corrupted with noise as it travels through the medium.

- Amplitude Shift Key

    In the Amplitude Shift Key modulation technique the amplitude of the sinusoidal is changed in response to a binary number for example zero or one. In such a modulation scheme the binary digit one could be represented by one amplitude of the sinusoidal and the binary digit zero could be represented by another amplitude. Below in figure 2.3 below shows a baseband signal of ones and zeros,

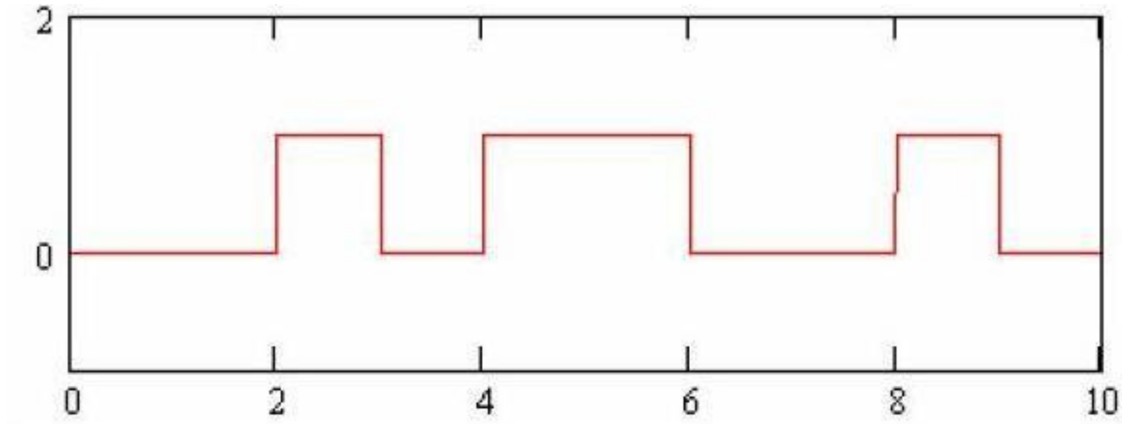


Figure 2.3 Baseband modulated signal of the sequence 0010110010 [20].

Figure 2.3 shows a baseband signal of a sequence 0010110010 this sequence is modulated using amplitude shift key in figure 1.4 and the equation for amplitude shift key is given below in equation 2.2,

$$ASK(t) = s(t)sin(2\pi ft) \quad \text{(Equation: 2.2)}$$

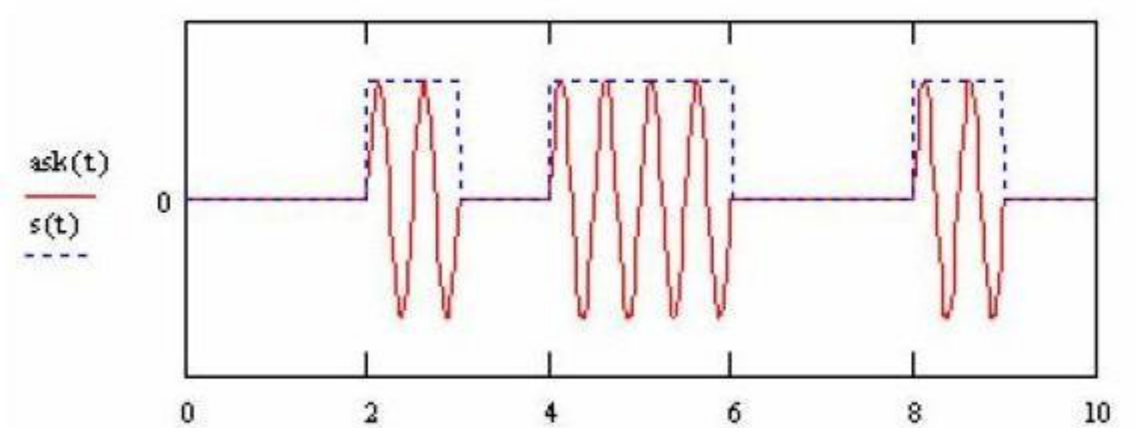


Figure 2.4 ASK modulation of baseband signal 0010110010 [20].

In figure 2.4 the amplitude for binary digit one is s(t) while the amplitude for binary digit zero is zero. This type of ASK is a special form of ASK called On-Off keying. Groups of binary digits can be represented by multiple amplitudes in the signal for different pairs of binary digits for example 00 could be represented by a signal with zero amplitude, 01 could be represented by a signal with n amplitude, 10 could be represented by a signal with n+1 amplitude and 11 could be represented by a signal with n+2 amplitude.

- Frequency Shift Keying:

In frequency shift keying modulation technique the frequency of the sinusoidal is changed in response to a binary digit in a baseband signal. That is in the case of just two binary digits 0 and 1 the binary 0 is represented by one frequency while the binary digit 1 is represented by another frequency. Figure 2.5 below shows the frequency shift key modulation of figure 2.3 and equation 2.3 is for FSK,

$FSK(t) = sin(2\pi f_1 t)\ for\ bit\ 1\ and\ FSK(t) = sin(2\pi f_2 t)\ for\ bit\ 0$ (Equation: 2.3)

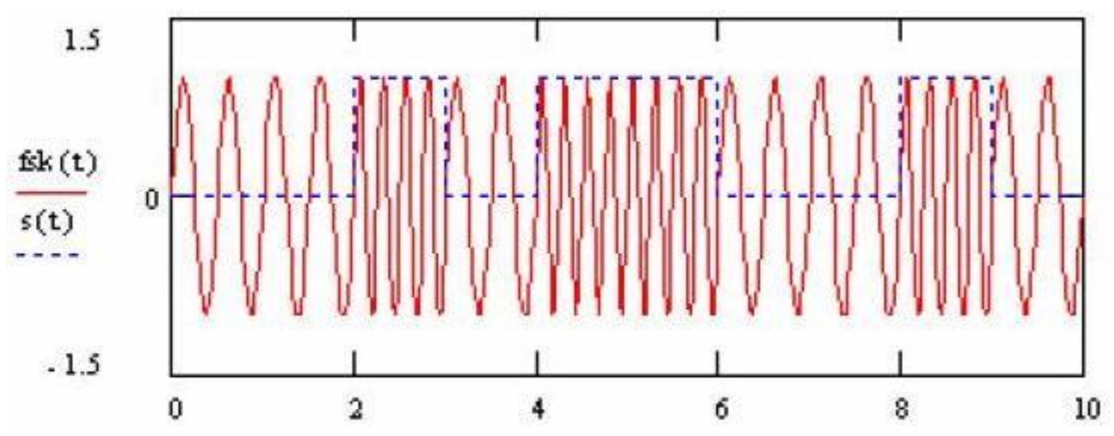


Figure 2.5 Frequency Shift key modulation of baseband signal 0010110010 [20].

As we can see from figure 2.5 that the binary zero digit is represented by one frequency and the binary one digit is represented by another digit.

- Phase Shift Keying:

In phase shift keying (PSK) modulation technique the phase of the signal is changed in response to a binary digit in a baseband signal. That is in the case of just two binary digits one and zero the zero is represented by no change in phase at the beginning of the sin wave and a one is represented by a 180 degree change at the beginning of the sin wave. Figure 2.6 shows the phase shift key modulation for the baseband signal in figure 2.3 and equation 2.4 gives is for PSK (actually BPSK),

$PSK(t) = sin(2\pi ft)\ for\ bit\ 1\ and\ sin(2\pi ft + \pi)\ for\ bit\ 0$ (Equation: 2.4)

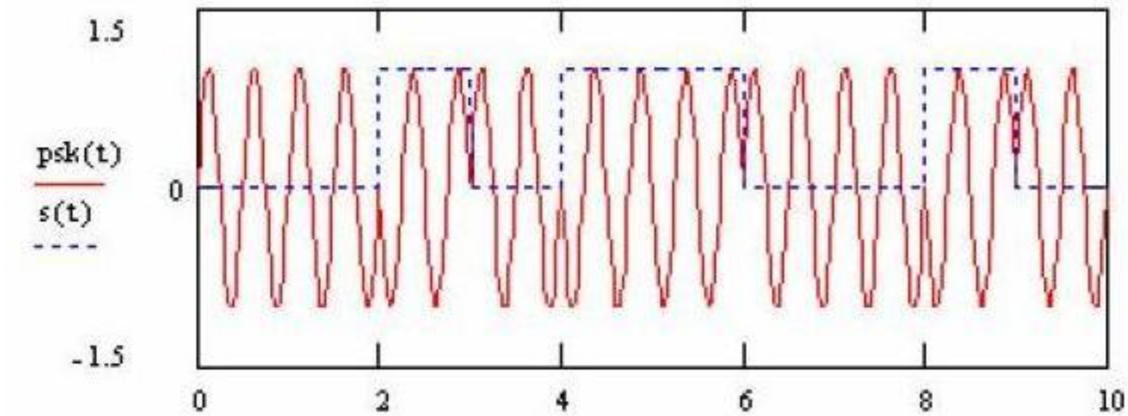


Figure 2.6 Phase shift key modulation of the baseband signal 0010110010 [20].

From figure 2.6 we can see that the signal is modulated by the phase and the starting angle of the phase determines the binary digit. The phase shift key modulation shown figure 2.6 is an example of binary phase shift key (BPSK). There are advance forms of PSK that are Quadrature Phase Shift Keying (QPSK), MPSK where M stands for the order of the modulations whether it is 2, 4 or 8. QPSK is a form of PSK in which the modulation is of 4 that is at the starting of a sin wave there are four different angles 0, 90, 180 and 270 which could represent binary digits 00, 01, 10, 11. It should be noted that both BPSK and QPSK are cases of MPSK where M=2 and 4. The equation for MPSK is given below,

$$S_i(t) = A_c p_c(t) cos\left(2\pi f_c t + \frac{2\pi i}{M}\right) \quad \text{(Equation: 2.5)}$$

In the above equation $p_s(t)$ is used for shaping the pulse that is as a pulse shaping function and f_c is the carrier frequency. In equation 6 the value of M selects the type of PSK and denotes the order of modulation that is binary PSK (M=2), quadrature PSK (M=4), 8PSK (M=8) and so on. There are several advantages of M-ary modulation over other types of modulation for example with QPSK we can transmit twice the amount of information that we could with BPSK in the same bandwidth. M-ary modulation schemes with M>8 requires complex circuitry and its recovery at the receiving end is also more complicated as compared to BPSK, therefore it is rarely used in practical situations.

In order to reduce the bandwidth we use pulse shaping functions as it is not practical to send square pulses because they require a lot of bandwidth and are hard to create. Therefore instead of transmitting square pulses we transmit pulses with different shapes that require a lesser amount of bandwidth and convey the same information. Such pulses also have many other characteristics like immunity towards intersymbol interference. Some of the techniques through which these pulses are shaped are given below,

- Root Raised Cosine (used in QPSK)
- Half-sinusoid (used in MSK)
- Gaussian (used in GMSK)
- Quadrature Partial Response(used in QPR)

In this thesis the method used for modulation is Gaussian Minimum Shift Keying (GMSK). In Minimum Shift Keying (MSK) a half sinusoid is used instead of a square pulse as a root raised cosine pulse is used instead of a square pulse in OQPSK. In GMSK a Gaussian pulse is used instead of a root raised cosine pulse or a half sinusoid. The Gaussian shaped pulse is represented by the formula,

$$g(t) = \frac{1}{2T}\left(Q\left(2\pi B_b \frac{t-.5T}{\sqrt{ln\,2}}\right) - Q\left(2\pi B_b \frac{t+.5T}{\sqrt{ln\,2}}\right)\right) \qquad \text{(Equation: 2.6)}$$

Where,

$$Q(t) = \int_t \frac{1}{\sqrt{2}} e^{-\left(x^2/2\right)} dx \qquad \text{(Equation: 2.7)}$$

In the above equation 2.6 the value Bb represents the bandwidth of the signal. The BT factor in equation 2.6 determines how sharply the Gaussian pulse rolls off. BT of 0.3 is usually used in practice. A GMSK modulated carrier has more spectral efficiency as compared to MSK Carrier. Also GMSK can be amplified by the use of non linear amplifier and remain undistorted. Non linear amplifiers have better power saving properties due to which they are preferred to be used in cell phones and other telecommunications equipment that is run on batteries. Also since noise is mainly amplitude based and GMSK does not use amplitude variations to convey information therefore GMSK is immune to different types of noise that are mainly amplitude based. GMSK is used by several cellular standards around the world such as Global Systems Mobile (GSM), Digital European Cordless Telephone (DECT). One disadvantage of GMSK is that it needs a higher level of power than QPSK and many other modulation schemes.

2.2.5 Fourier Transform

In 1807, Jean Baptiste Joseph Fourier showed that any periodic signal could be represented by a series of sinusoidal functions. The Fourier series was used in order to convert periodic signals from the time domain into the frequency domain. Fourier transform is a mathematical transform that is used to convert aperiodic signal that is in the time domain into the frequency domain. This is necessary in order to be able to view individual frequencies which are not visible in the time domain. The Fourier Transform of a aperiodic signal x(t) shown in figure 2.7 shown below,

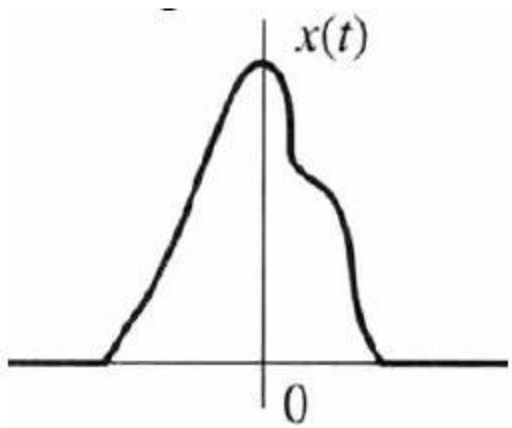


Figure 2.7 An aperiodic signal X(t)

can be obtained by Fourier Transform equation: 2.8 below,

$$X(\omega) = \mathcal{F}[x(t)] = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt \qquad \text{(Equation: 2.8)}$$

In the above equation 2.8 x(t) is the original function which is in the time domain and X(w) is the Fourier Transform of the function which is in the frequency domain. The Inverse Fourier transform is given in equation: 2.9 below,

$$x(t) = \mathcal{F}^{-1}[X(\omega)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}\,d\omega \qquad \text{(Equation: 2.9)}$$

This equation is used to convert the signal from the frequency domain back into the time domain. In this thesis the Fourier transform is used to verify the different individual frequencies produced by the frequency hopping sequence. A major advantage of the Fourier Transform is that they convert a function from one domain into another domain without the loss of any information.

But as computers work with digital data only thus we use the Discrete Fourier Transform DFT. The Discrete Fourier Transform (DFT) is different from the Discrete Time Fourier Transform (DTFT), the Discrete Time Fourier Transform takes as an input a discrete signal in the time domain and returns a continuous signal in the frequency domain.

Whereas the Discrete Time Transform takes a discrete time signal as an input and returns a discrete signal in the frequency domain. This implies that the basic difference is that the DTFT is continuous in the frequency domain while the DFT is discrete in the frequency domain. The DFT can be obtained by the following equation 2.10 below,

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \qquad j = 0, \dots, n-1 \qquad \text{(Equation: 2.10)}$$

Due to the DFT we are able to compute the Fourier Transform on a computer. The Fast Fourier Transform is a computationally efficient method of implementing the DFT in a much shorter amount of time. The Fast Fourier Transform achieves this efficiency by reducing the number of multiplications required to obtain the desired result from N^2 to $N/2\log_2 N$. Many techniques were introduced in order to reduce the computational time and complexity but the most popular is the Cooley-Tukey Algorithm which is a radix-2 decimation in time FFT. The Cooley-Tukey works by reducing a DFT of size N recursively into smaller parts. One of the most common methods is to divide the transform into a size of N/2, which implies length equal to the power of 2 is required. Due to this reason there are cases in which sequences that are not of length equal to the power of 2 needs to be padded with zeros in order to make their length equal to 2. FFT are used in digital signal processing systems from everything in telecommunications equipment to image processing systems.

Proposed Technique

Chapter 3

3.1 Overview

In this Thesis the proposed technique for frequency hopping in a staggered multi-hop multi-subnet environment is accomplished through producing a staggered hoping sequence for multiple access while also in parallel producing a TDMA superframe for multi-hoped multi-subnet environment in synchronization with the staggered hopping sequence for multiple access. Once the subnets are synchronized, all of the radios among each subnet have their own hopping pattern. Although in this thesis this technique has been implemented for 30 radios and for a subnet of four radios supporting two hops between the source and the destination radio it can be further extended to n hops. The TDMA super frame is used initially for synchronizing all of the communicating radios in which each radio sends requests and acknowledgements in their own specified time slots as to which radio wants to communicate with which radio. Similarly all radios update their respective tables as to which radio has requested which radios to communicate with also acknowledgements are stored in these tables. During this portion of the TDMA superframe all of the radios are hopping on the same frequency hopping pattern. After that in the second portion of the TDMA superframe the data communication part starts. In this portion of the TDMA superframe the data is communicated between the source radio, hop 1 radio, hop 2 radio and destination radio. Due to the popularity and extensive amount of research that is being carried out on multi-hoping, new algorithms are being proposed on all layers of the ISO stack. This technique is implemented on the Datalink Layer (Layer 2) of the ISO stack.

3.2 Staggered hopping for multiple access

Linear Feedback Shift Register (LFSR) is used in order to generate a seemingly random sequence for a frequency hopping pattern. The reason for this being that the sequence of an m-bit LFSR is repeated after 2^m-1 outputs. Therefore by using of a 32 bit LFSR the sequence was being repeated after 2^{32}-1=4294967295 outputs. Due to the fact that the LFSR's output is then hashed detecting such a lengthy sequence is difficult and it appears random. A Linear Feedback Shift Register is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Therefore a LFSR is most commonly designed as a shift register whose input which is its input bit is the the exclusive-or (XOR) function of some bits of the overall shift register value. The values of a Linear Feedback Shift Register are deterministic and must be repeated after a cycle but by choosing a large number of bits for the Linear Feedback Shift Register can produce a sequence of bits which appears random and which has a very long cycle. In the technique used in this paper a 32-bit Linear Feedback Shift Register was used. The figure 3.1 shows the 32 bit LFSR that was used in this paper,

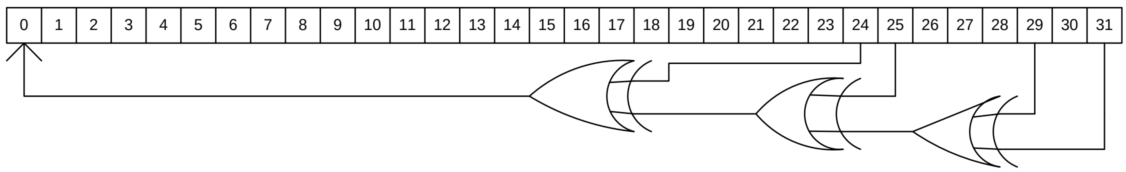


Figure 3.1 32-bit Linear Feedback Shift Register.

In the LFSR used in this thesis the registers 31, 29, 25 and 24 were used to feedback the values of the shift register into the input bit of the shift register.

Hash Functions are any functions or algorithms that map larger datasets to smaller datasets and the smaller datasets are called hashed outputs. Hash Function is used in this technique in order to map the 32-bit output of the LFSR to an 8 bit number. Therefore the total range of the 8 bit number is from 0 to 255.This implies that the technique used in this paper can support a total of 256 frequency slots through which the frequency hopping sequence can occur. This implies that a total of 256 radios can be supported by this technique as a total of 256 frequencies can be generated in parallel, although in our simulations which we conducted to test this technique, we have only used 30 radios and generated 30 frequencies in parallel out of which one frequency hopping slot was allocated to each radio.

The Hash Function that was used in this paper is the Pearson Hash Function that is given in [2].The Pearson Hashing Function is a quick, simple and efficient hash function that hashes any number of bytes in to 8 bit numbers it is ideal for systems with limited resources and for power efficiency. The algorithm is,

```
n := 0

for each c in C loop

  index := n xor c

  n := T[index]

end loop

return n
```

In the above algorithm T is a 1-dimensional array containing pre-calculated values of Pseudorandom Permutation for the Integers 0 through 255. According to [2] this arrangement of numbers from 0 to 255 gives good hashing behavior. The table of the pre-calculated values of Pseudorandom Permutation for the Integers 0 through 255 that is used in this technique from [2] is given below in figure 3.2,

1	87	49	12	176	178	102	166	121	193	6	84	249	230	44	163
14	197	213	181	161	85	218	80	64	239	24	226	236	142	38	200
110	177	104	103	141	253	255	50	77	101	81	18	45	96	31	222
25	107	190	70	86	237	240	34	72	242	20	214	244	227	149	235
97	234	57	22	60	250	82	175	208	5	127	199	111	62	135	248
174	169	211	58	66	154	106	195	245	171	17	187	182	179	0	243
132	56	148	75	128	133	158	100	130	126	91	13	153	246	216	219
119	68	223	78	83	88	201	99	122	11	92	32	136	114	52	10
138	30	48	183	156	35	61	26	143	74	251	94	129	162	63	152
170	7	115	167	241	206	3	150	55	59	151	220	90	53	23	131
125	173	15	238	79	95	89	16	105	137	225	224	217	160	37	123
118	73	2	157	46	116	9	145	134	228	207	212	202	215	69	229
27	188	67	124	168	252	42	4	29	108	21	247	19	205	39	203
233	40	186	147	198	192	155	33	164	191	98	204	165	180	117	76
140	36	210	172	41	54	159	8	185	232	113	196	231	47	146	120
51	65	28	144	254	221	93	189	194	139	112	43	71	109	184	209

Figure 3.2 Pseudorandom Permutations of the Integers 0 through 255 [8].

The Hashed output is then used as an index for the frequency table which is actually an array or ROM containing the pre-defined frequencies. The problem with hashing a 32 bit number to an 8 bit number is that some numbers are at certain times repeated. Such circumstances can lead to collision between frequencies which in effect could lead erroneous communications. This is due to the fact that in such a situation two frequency slots would be allocated the same frequency which in effect would cause the erroneous communications among multiple radios if they were using this frequency hopping technique for multiple accesses.

In order to avoid such circumstances we first store the frequencies to be allocated in to an array before allocating them. In our simulations as we were testing the technique for 30 radios, therefore the array size is kept at 30. The first frequency stored in the array is not compared with any previously stored frequency but the subsequent frequencies are compared with the previously stored frequencies. If there is a match found then that hashed output is discarded and not stored in the array. Then a new or the next hashed output is generated this again goes through the array and is compared with any previous entries in the array. This process goes on until the array is full. Once the array is full the frequency indexes are allocated in parallel. These indexes are further used to select the corresponding frequency for transmission. This could be done through a ROM in hardware.

For transmitting the data we used Gaussian Mean Shift Key (GMSK) modulation. Gaussian Mean Shift Key is a very popular modulation scheme and is used in several wireless communication systems around the world. Some of the wireless communication systems that use GMSK are Cellular Digital Packet Data (CDPD), Digital European Cordless Telephone (DECT), Global Systems Mobile (GSM), both PCS1900 and DCS1800 for GSM cellular systems also uses GMSK in Europe and America respectively. Unlike Mean Shift Key (MSK) in which we send half cycle sinusoidal as a pulse we send a Gaussian shaped pulse. The main problematic issue with square pulses is that creating a square pulse requires much more bandwidth and is more difficult as compared to creating a Gaussian pulse that's why they are not practically used. So instead of sending square pulses Gaussian shaped pulses are sent and they convey the same meaning while at the same time require a smaller amount of bandwidth. In the process of pulse shaping there is a parameter called the roll-off which controls the shape and bandwidth of the signal. In GMSK we use a Gaussian shaped pulse which reduces the amount of bandwidth even more as compared to Mean Shift Key (MSK). The Gaussian shaped pulse was given in equations 2.6 and 2.7,

3.3 Hopping Sequence Analysis

The bit stream that we used was NRZ 11111111 and then it was GMSK modulated and the index chooses the frequency which is used to further generate the sinusoidal. Both the modulated GMSK and the sinusoidal were resampled and multiplied. By taking and plotting the absolute of the Fourier transform in the frequency domain we can see that the first frequency of 1390Hz has value at 447 radians/sample. The second frequency of 1950Hz has a value at 404 radians/sample. The third frequency of 1800Hz has a value at 58 radians/sample. The fourth frequency of 2080Hz has a value at 363 radians/sample. The fifth frequency of 790Hz has a value at 256 radians/sample. The sixth frequency of 960Hz has a value at 303 radians/sample. The seventh frequency of 660Hz has a value at 211 radians/sample. The eighth frequency of 1080Hz has a value at 345 radians/sample. The ninth frequency of 2030Hz has a value at 379 radians/sample. The tenth frequency of

1360Hz has a value at 434 radians/sample. Figure 3 below shows the results from the simulations that were performed.

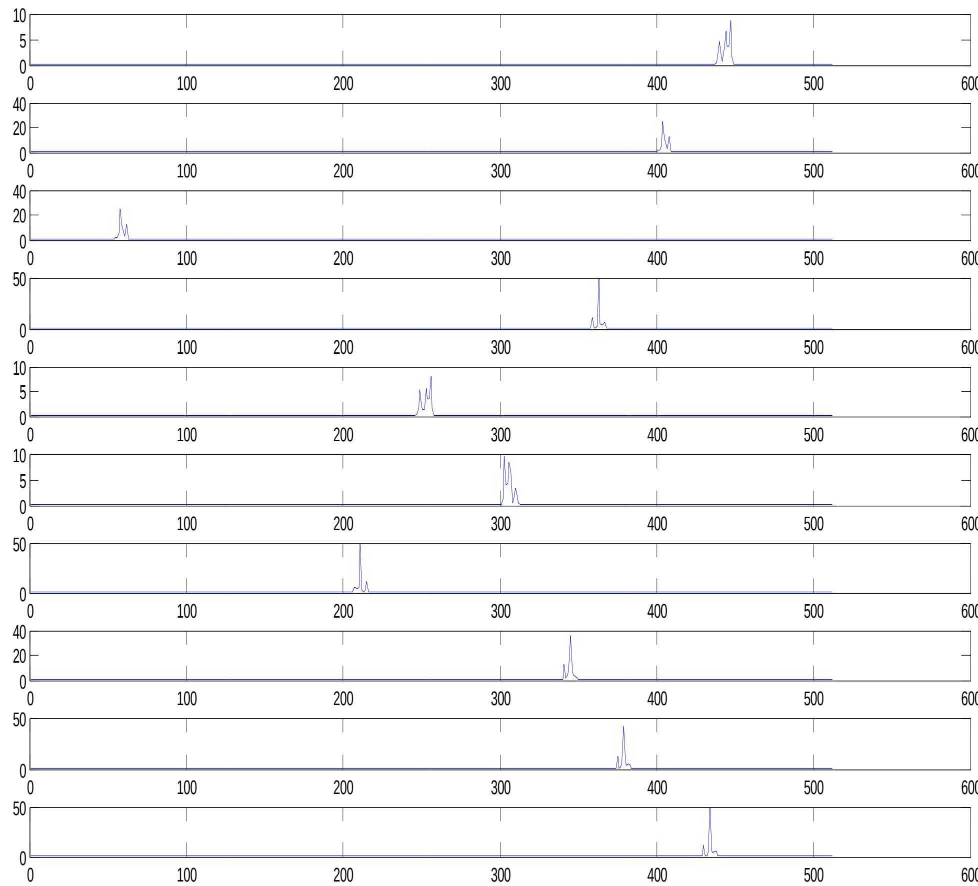


Figure 3.3 Ten frequencies generated in parallel for multiple access.

Figure 3.3, Frequency Domain representation of the different allocated frequencies for first 10 frequency slots. From the above analysis we can see that the frequencies are properly being allocated to different frequency slots.

3.4 TDMA Super frame for Multi-hopped Multi-Subnet

The TDMA super frame that was used in order to synchronize different radios among each other and then create multi-hopped multi-subnets has a total length of 9600ns, each time slot is of 20ns. The first part of the super frame is for synchronization purposes and the second portion is for data transfer. Initially all of the 30 radios are hopping on the same frequency hopping pattern for the synchronization part of the super frame. The structure of the super frame is such that the first 30 time slots are for each of the radios (in this case source radios) to transmit the address of the respective destination radios that they want to communicate with during their respective time slot allocated to them. During this specific time all of the other radios will update their respective tables as to which radio has requested to communicate with which radio as a destination radio. This is for time slots from 0 to 29. After that the next 30 time slots are for each of the radios to send a request for hop 1 that is for a radio that the requesting radio would want it to work as hop 1 for it. It should be noted that this implementation is working on the ISO Datalink Layer (Layer 2) and that the request for destination, hop 1, hop 2 radios will be coming from the network layer. Similarly during these 30 time slots all of the radios in the network will update their own tables as to which radio requested which radio to work as hop 1.In the next 30 time slots the source radio request Hop 2 (that is the radio it wants it to work as Hop 2 for it) in its own time slot. Similarly all the corresponding radios in the network will update their tables as to which radio was requested as Hop 2. Then in the next 30 time slots the radio that was requested to work as Hop 1 will transmit an acknowledge during its own time slot. Similarly all of the radios will update their respective tables as to which radio sent an acknowledge and which did not send an acknowledge. If the radio that was requested to work as Hop 1 sends an acknowledge then in the next 30 time slots than any radio that has sent an acknowledge to work as Hop 1 for a source radio, will send a request to the radio that was requested to work as Hop 2 by that same source radio in its own time slot. Again similarly all of the radios will updated their respective tables as to which radio working as Hop 1 requested which radio to work as Hop 2. Then in the next 30 time slots any radio requested to work as Hop 2 will transmit an acknowledge if it wants to during its own time slot. Again all of the radios will update their respective tables as to which radio transmitted an acknowledge to work as Hop 2. Then in the next 30 time slots any radio that transmitted an acknowledge to work as Hop 2 will send a request to the destination radio that was requested by the source radio. During these 30 time slots all of the radios will again update their respective tables as to which radio sent a request to which destination radio also during which time slot of the super frame. Then if the destination radio wants to communicate with the source radio it will transmit an acknowledge to the hop 2 radio during the next 30 time slots in its own specified time slot. All of the other radios will update their tables respectively as they do after each 30 time slots of the super frame.

During the next 60 time slots Hop 2 will transmit a acknowledge to Hop 1 after receiving an acknowledge from the destination radio, in the first 30 time slots then after that during the next 30 time slots Hop 1 will transmit an acknowledge to the source radio. During these 60 time slots all of the radios will correspondingly update their specific tables as well. Up till this portion of the super frame all of the radios have the same frequency hopping pattern and the frequency hops to a new frequency after every 30 timeslots. As soon as these 60 time slots are complete the frequency hopping pattern of the four radios that is source, hop 1, hop 2 and destination radio in a specific subnet are transferred to one of the hopping patterns among the 30 that are generated in parallel from the above staggered hopping for multiple access technique given above. The radios that are in a subnet are transferred to the hopping pattern of the source radio index. That is if the source radio is ‘0’ then the hopping pattern is the ‘0’ frequency hopping pattern generated in parallel among the 30 frequencies generated in parallel. Also the frequency hops after every 30 time slots just as it did for the synchronization part of the super frame. This is due to the fact that the data transfer part starts now.

Now in the next 30 time slots the source radio starts to transmit data to hop 1 in this case we transmitted a certain number of bytes from the source radio to the hop 1 radio after verifying the second acknowledge from the hop 1 radio. Similarly in the next 30 time slots the hop 1 radio retransmits the data in its own specified time slot after verifying the second acknowledge from hop 2, then after receiving the data hop 2 retransmits the data from hop 2 to the destination radio in its specified time slot in the next 30 time slots after verifying the acknowledge from the destination radio. After the destination radio receives data in the next 30 time slots it transmits data in its specified time slot. The data transmitted from the destination radio is received at hop 2 radio and in the next 30 time slots it is retransmitted from the hop 2 radio to the hop 1 radio during the timeslot of hop 2 radio after verifying the destination acknowledge that was sent previously. Similarly the data is then received at the hop 1 radio and then retransmitted during the next 30 time slots from the hop 1 radio to the source radio after verifying the hop 2 second acknowledge. After receiving the data at the source radio the super frame starts again and all the radios are shifted back on to the same frequency hopping pattern for synchronization.

Implementation and Experimental Results

Chapter 4

In this chapter we have implemented the proposed technique for staggered frequency hopping in a multi-hopping multi-subnet environment. The proposed technique was initially implemented in MATLAB after its implementation and verification in MATLAB the proposed technique was designed in Verilog and simulated in ModelSIM SE 5.7f. The test data is given which was used for both the MATLAB simulations and Verilog simulations.

4.1 Experimental Environment

The following hardware and software specifications that were used in our simulations are listed below,

4.1.1 Hardware

In our simulations we used a Laptop with the following specifications Core 2 Duo 1.6 GHz, 1GB RAM and 160 GB Hard Drive.

4.1.2 Software

The following are the specifications of the software's used in this thesis,

MATLAB R2010a is a industry standard software package that is used for communications simulations and also for simulating various other engineering systems. MATLAB has become the tool of choice for many engineering simulations because of its richness in libraries and graphical tools. Once the code is tested in MATLAB it is then translated into VERILOG. MATLAB was used for the initial simulations of the staggered hopping for multi-hopping multi-subnet technique.

ModelSim SE 6.3f was used for simulation of verilog code. ModelSim SE 6.3f is an excellent simulation tool. It also has the functionality to import libraries from other vendors like Altera, Actel and Xilinx.

4.2 MATLAB Simulations

The staggered hopping in a multi-hop multi-subnet environment was first implemented and tested in MATLAB. The MATLAB code supported up to a total of 10 radios but can be extended to n radios. Four radios are allocated per subnet supporting two hops between the source and destination radio. During the initial 10 timeslots all of the radios request the destination radio that they want to communicate with in their own specified timeslots. Also as each radio sends its request the other radios update their tables respectively. In our MATLAB simulation if a source radio does not want to request any destination radio then it transmits a 0. Also during the initial 10 time slots all of the radios are transmitting their requests on the same frequency. The frequency hops after 10 time slots to another frequency. Although in this MATLAB simulation we are testing the proposed technique for 10 radios we have generated up to 30 frequencies in parallel also although 30 frequencies are generated in parallel we are only using 10 frequency slots as there are only 10 radios. Similarly just as the number of radios can be extended to n radios, the number of frequencies that can be generated in parallel for multiple access can be extended to n frequencies up to a total of 256 frequencies as the hashed output of the LFSR is an 8 bit number. In the beginning of the run of the MATLAB simulation the program first generates the 30 frequencies in parallel and allocates the frequency in the first slot to all of the radios during the first 10 time slots of the synchronization and subnet creation part of the super frame. As we can see that below in figure 4.1 in the variable editor which allows us to view the contents of a variable the values of the 30 frequencies generated in parallel.

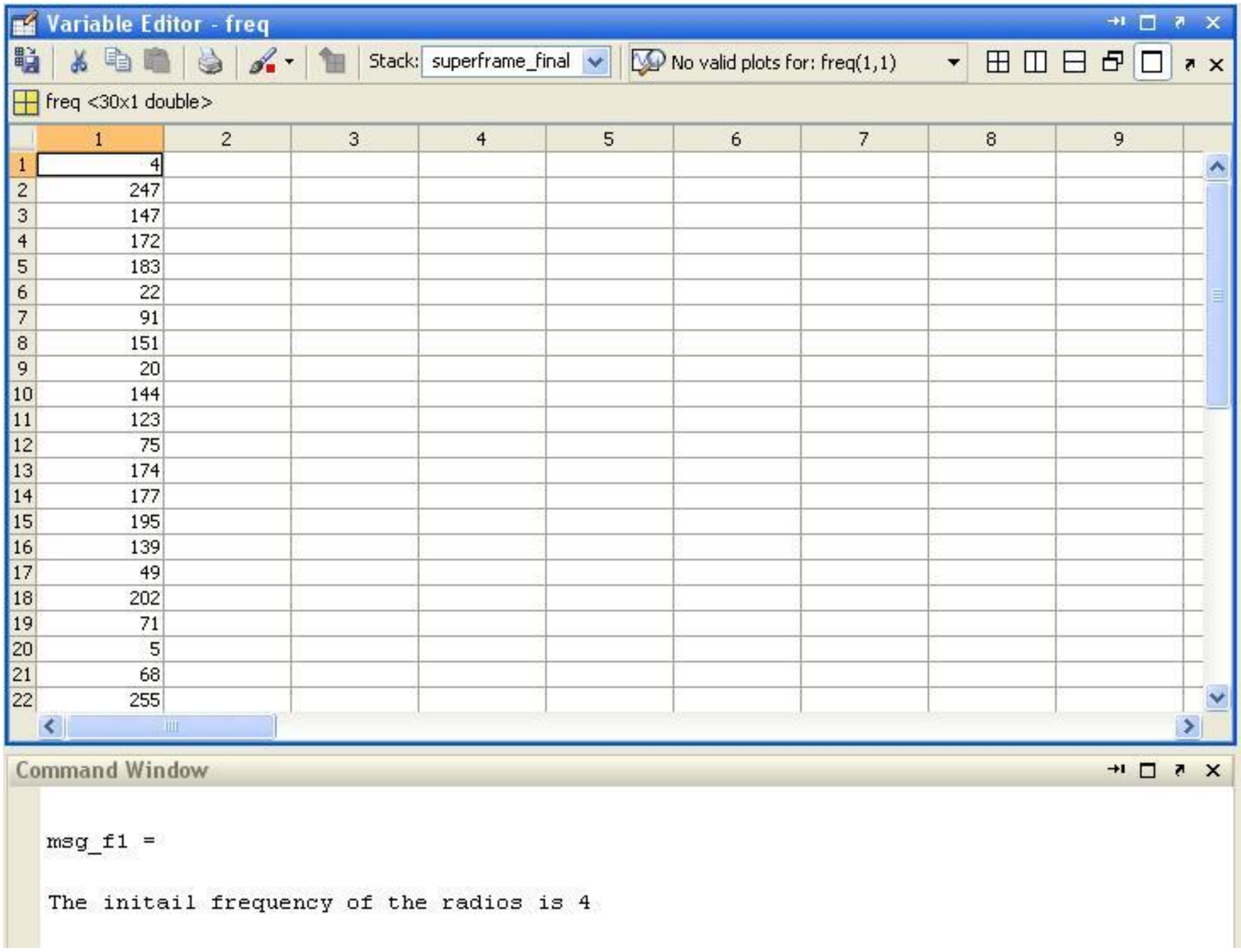


Figure 4.1 It shows the 30 frequencies generated in parallel.

In the above figure we can see that the random frequency allocated to the first frequency slot is used by all the radios during the first 10 time slots in which the radios that want to act as a source radio send requests for a destination radio. After this all of the radios that are acting as source radios send requests for their destination radio in their respective time slots the first time slot is for radio 1 and second time slot is for radio 2 and so on.

In our MATLAB simulation for the first subnet the first source radio is radio 1 which requests radio 10 as its destination radio, radio 5 as hop 1 radio and radio 8 as hop 2 radio. For the second subnet the source radio is radio 2 which requests radio 4 as its destination radio, radio 7 as hop 1 radio and radio 3 as hop 2 radio. As we can see in from figure 4.2 below the during the first 10 time slots the source radios request their destination radios. From figure 4.2 we can see that in the first time slot radio 1 has requested radio 10 as its destination radio and the source radio 2 has requested radio 4 as its destination radio. Since the rest of the radios are not requesting any other radio they simply transmit zero.

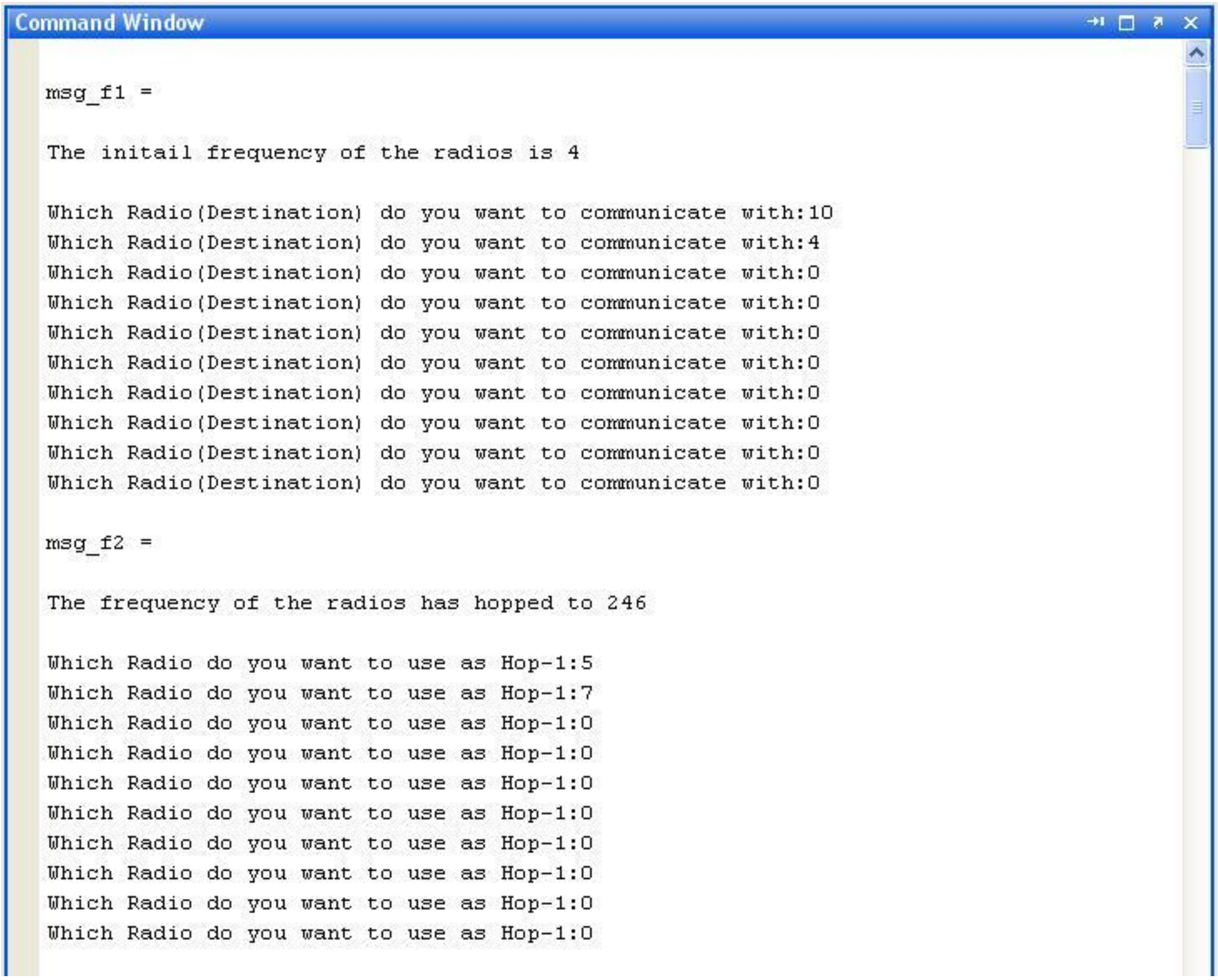


Figure 4.2 First 20 time slots of the super frame for destination and hop 1 request

In the above figure 4.2 we can see that in the first time slot which is for the radio 1, radio 1 requests radio 10 as its destination radio and in the second timeslot radio 2 requests radio 4 as its destination radio. After the first 10 time slots the frequency on which all of the radios are communicating hops to another frequency and after that in the first time slot of the next 10 time slots radio 1 requests radio 5 to work as its hop 1 radio and radio 2 requests radio 7 to work as its hop 1 radio. All of the other radios update their respective tables accordingly as to which radio requested which radio to work as a destination radio and hop 1 radio. As the other radios are not requesting any other radio as a destination radio or a hop 1 radio they simply transmit zero which is means no request in our MATLAB simulations. At the start of the next 10 time slots the frequency hops to another frequency in the first frequency slot of the 30 frequencies of 30 slots that are generated in parallel which is shown below in figure 4.3,

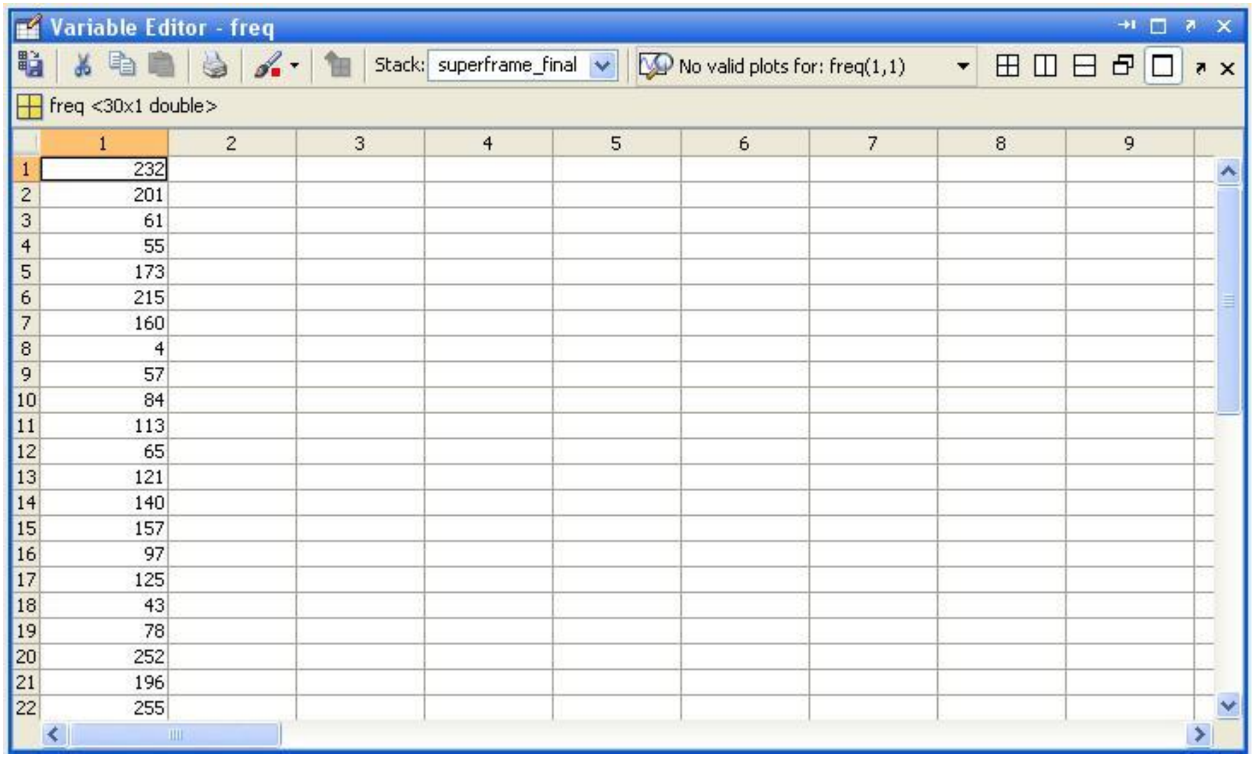


Figure 4.3 Frequency generated in parallel in the 30 frequency slots.

Similarly in our next 10 timeslots the source radios 1 and 2 request radios 8 and 3 as its hop 2 radios. This is shown in the below figure 4.4,

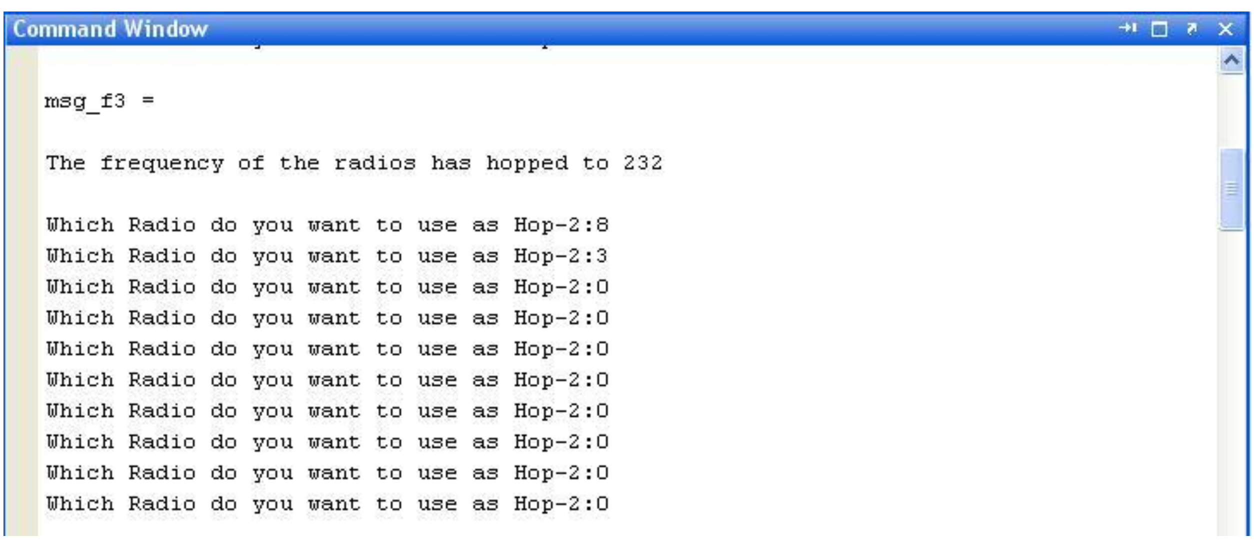


Figure 4.4 Source radios requesting Hop 2 radio

Similarly the frequency hops to another frequency at the end of the 10 time slots and in the next 10 time slots the radio that was requested to work as hop 1 transmits its acknowledge in its specified time slot. In our MATLAB simulations if a radio wants to acknowledge then it transmits a one otherwise it transmits a zero. This is shown in figure 4.5,

```
Command Window

msg_f4 =

The frequency of the radios has hopped to 171


msg1 =

Do you want to work as a Hop-1 for 0

Acknowledge with 1 reject with 0:0

msg1 =

Do you want to work as a Hop-1 for 0

Acknowledge with 1 reject with 0:0

msg1 =

Do you want to work as a Hop-1 for 0

Acknowledge with 1 reject with 0:0

msg1 =

Do you want to work as a Hop-1 for 0

Acknowledge with 1 reject with 0:0

msg1 =

Do you want to work as a Hop-1 for 1

fx Acknowledge with 1 reject with 0:1
```

Figure 4.5 Hop 1 Acknowledge by radios requested to work as Hop 1 first five time slots.

In the above figure 4.5 we can see that in the first four time slots no radio transmits acknowledge as none of the first four radios was requested to work as a hop 1 by any of the source radios 1 or 2. But in the fifth time slot the fifth radio transmits acknowledge during its own time slots and all of the radios update their respective tables accordingly. Also in figure 4.6 given below we can see that in the seventh time slot the radio 7 transmits an acknowledge to work as hop 1 radio for the source radio 2 and the rest of the radios transmit a zero as they have not been requested to work as a hop 1 radio by any source radio.

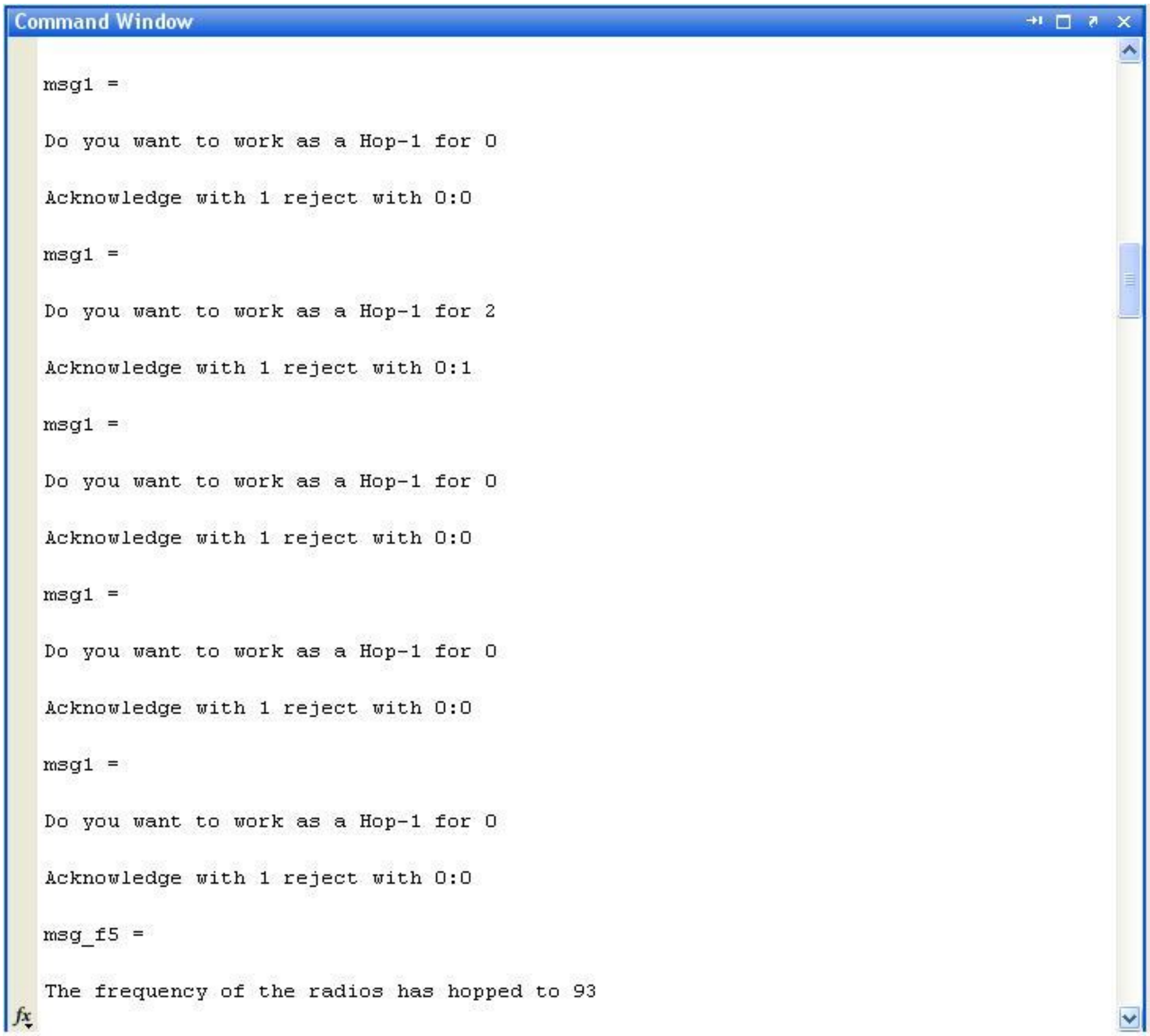


Figure 4.6 Hop 1 Acknowledge by radios requested to work as Hop 1 last five time slots.

After these 10 time slots the frequency hops to another frequency which is 93 as can be seen from the figure 4.6. After these 10 time slots in the next 10 time slots the Hop-1 radio sends a request for the Hop-2 radio in its own specified time slot by transmitting the number of the radio. As we can see below in figure 4.7 in the fifth time slot the hop-1 radio 5 requests radio 8 to work as its hop-2 radio and in the seventh time slot the hop-1 radio 7 requests radio 3 to work as its hop-2 radio.

```
msg2 =

Request to communicate with Hop-2 Radio 8


msg2 =

Request to communicate with Hop-2 Radio 3


msg_f6 =

The frequency of the radios has hopped to 96
```

Figure 4.7 Hop-1 radios 5 and 7 request Hop-2 radios 8 and 3 in their own specified time slots.

After these 10 time slots are completed the frequency hops to another frequency. Similarly all of the radios update their tables as to which radio requested which radio to work as a hop-2 radio. Then in the next ten time slots of the first portion of the super frame the radios that were requested to work as a Hop-2 radio transmit an acknowledge if willing during their own specified time slots. In figure 4.8 below we can see that radio 3 which was requested to work as hop-2 radio for radio 7 transmits an acknowledgement in its own time slot and radio 8 which was requested to work as a hop-2 radio for radio 5 transmits an acknowledgement in its own time slot.

```
msg3 =

Do you want work as a Hop-2 for Radio 7

Acknowledge with 1 reject with 0:1

msg3 =

Do you want work as a Hop-2 for Radio 5

Acknowledge with 1 reject with 0:1

msg_f7 =

The frequency of the radios has hopped to 79
```

Figure 4.8 Hop-2 acknowledgments in its time slot to Hop-1 radios.

After these 10 time slots are completed the frequency hops to another frequency. In the next ten time slots the radios that were requested to work as hop-2 radios transmit requests to the radios that were requested to work as destination radios in their own specified time slots. As we can see in figure 4.9 below that radio 3 which is working as hop-2 radio for source radio 2 requests the destination radio 4 in its own specified time slot and radio 8 which is now working as the hop-2 radio for the source radio 1 requests to communicate with the destination radio 10 by transmitting the radio number in its own specified time slot,

```
Command Window

msg4 =

Request to communicate with destination Radio 4 from Radio 3


msg4 =

Request to communicate with destination Radio 10 from Radio 8


msg_f8 =

The frequency of the radios has hopped to 225
```

Figure 4.9 Hop-2 radio requests to communicate with destination radio.

The frequency hops to another frequency in the first frequency slot in the first 10 time slots. After that in the next ten time slots the destination radio transmits an acknowledgement if it is willing to communicate during its own specified time slot. This is shown in figure 4.10 below,

```
msg5 =

Do you want to communicate with Hop-2 Radio 3

Acknowledge with 1 reject with 0: 1

msg5 =

Do you want to communicate with Hop-2 Radio 8

Acknowledge with 1 reject with 0: 1

msg_f9 =

The frequency of the radios has hopped to 100
```

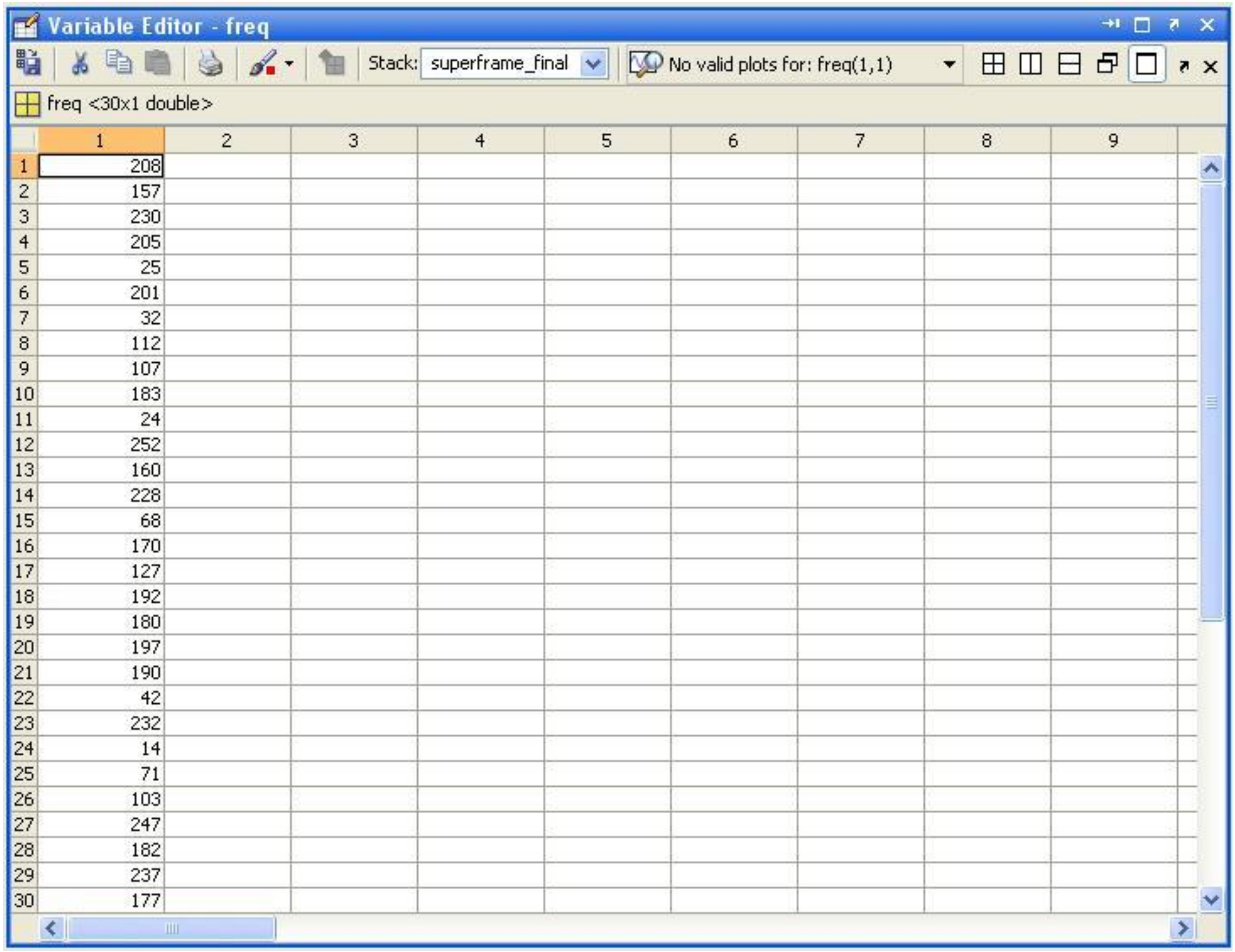
Figure 4.10 Destination Radio Acknowledgements

From the above figure 4.10 we can see that the destination radios 4 and 10 transmits an acknowledgement in its own specified time slots. Then after the 10 time slots are completed the frequency hops to another frequency. After that in the next 10 time slots the Hop-2 radios transmit an acknowledgement to hop-1 radios in their own specified time slots all of the radios correspondingly update their tables respectively. After that the frequency hops to another frequency and in the next 10 time slots the hop-1 radios transmit acknowledgements to the source radios.

After this first portion of the super frame which is for synchronization and creation of multi-subnets in a multi-hop environment in which all of the radios are hopping on the same frequency pattern. In the second portion of the super frame all of the four radios in each subnet that are a part of a subnet start hoping on their own staggered frequency hopping pattern. This was introduced so the each subnet would have its own hopping pattern for multiple access which would not interfere with the hopping patterns of the other subnets. This in effect improves and enhances security from eavesdropping and keeps communications between radios of a specific subnet independent of communications between radios of other subnets.

Now in the second portion of the super frame the radios of a specific subnet start to hop on the frequency pattern that is coming from the frequency slot number that is the same as that of the source radio number. That is if the source radio is 1 then out of the 30 frequency slots generating 30 hopping patterns in parallel the hopping pattern of the subnet which has radio 1 working as its source radio is obtained from the 1^{st} frequency slots among the 30 slots that are producing frequency hopping patterns in parallel. Therefore in our implementation in MATLAB the group of four radios in the first subnet starts hopping on the frequency pattern of the first frequency slot because the source radio is radio 1(that is radio number 1) and the group of radios in the second subnet start hopping on the frequency pattern of the second frequency slot because the source radio is radio 2.

In the figure 4.11 below we can see the first two frequencies that are allocated to the two subnets in the implementation.


Variable Editor - freq

Stack: superframe_final | No valid plots for: freq(1,1)

freq <30x1 double>

	1
1	208
2	157
3	230
4	205
5	25
6	201
7	32
8	112
9	107
10	183
11	24
12	252
13	160
14	228
15	68
16	170
17	127
18	192
19	180
20	197
21	190
22	42
23	232
24	14
25	71
26	103
27	247
28	182
29	237
30	177

Figure 4.11 Frequencies generated in parallel for allocation to subnets in first 10 time slots of second portion of the super frame

In the above figure 4.11 we can see the 30 frequencies that are generated in parallel the frequencies generated in the first two frequency slots are allocated to the two subnets. Figure 4.12 below shows the frequencies that are allocated to the two subnets and the transfer of data from the source radio 1 and 2 to the hop-1 radios 5 and 7 in the time slots of the source radios in the first 10 time slots of the second portion of the super frame. As we can see in the figure 4.12 below the frequencies in the first two frequency slots are allocated to the two subnets with source radios 1 and 2.

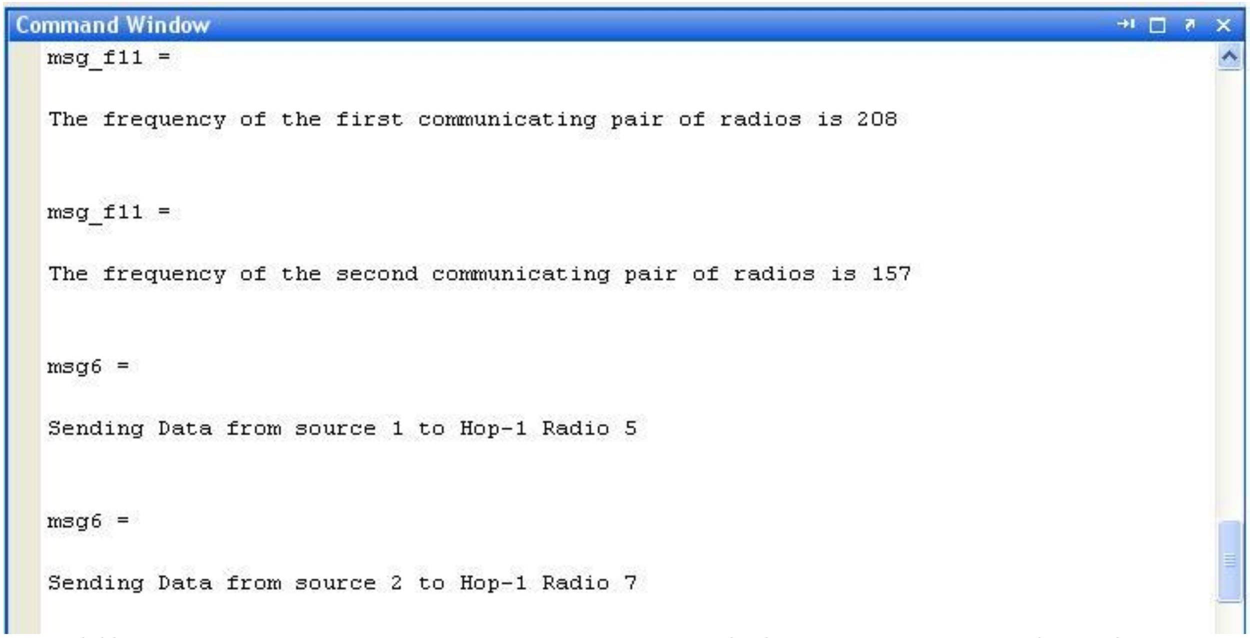


Figure 4.12 Frequencies allocated to the two subnets and data transfer from source radio to hop-1 radio for both subnets.

After these 10 time slots the frequencies on which the two subnets are hoping hops to another frequency. Also the data is transmitted from the hop-1 radios to the hop-2 radio in the time slots of the hop-1 radio. This is shown in figure 4.13 below,

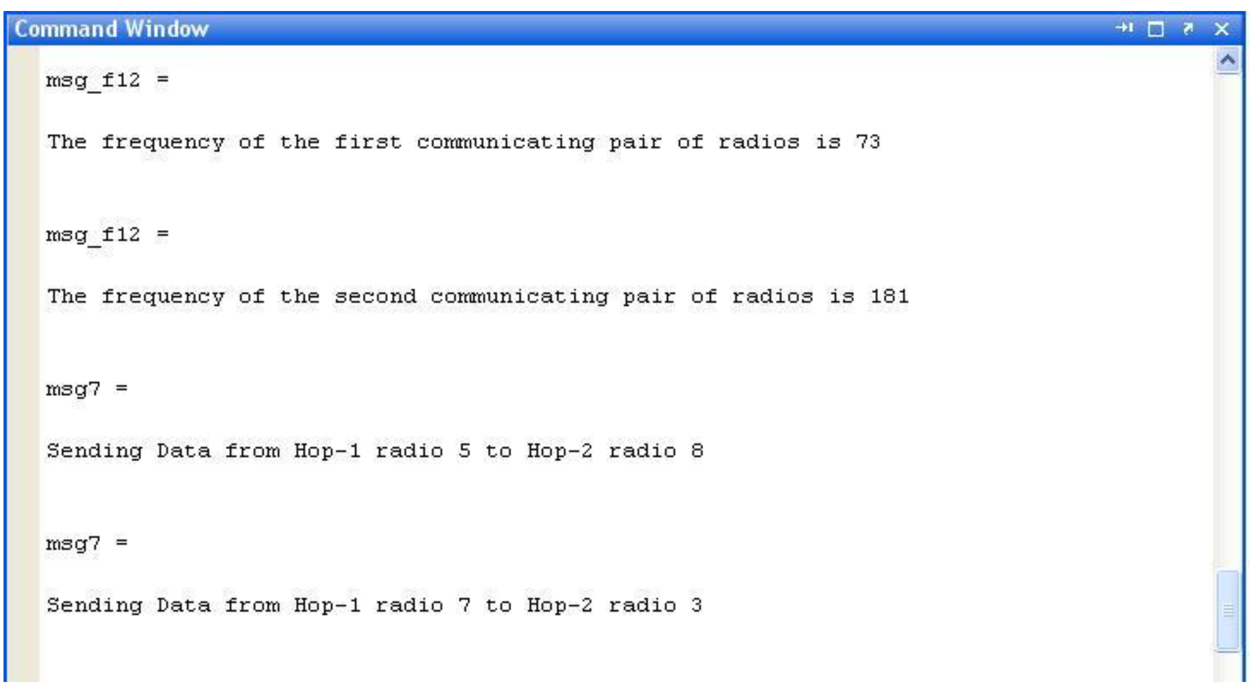


Figure 4.13 Frequencies allocated to the two subnets and data transfer from the hop-1 radio to the hop-2 radio for both subnets.

As we can see that the frequencies of the two subnets have hoped to other frequencies and the data is being transferred from the hop-1 radios to the hop-2 radios. After this in the next 10 time slots the frequencies in which the group of radios of the two subnets are communicating hops to another frequency and the data is transferred from the hop-2 radio to the destination radio. This is shown below in the figure 4.14,

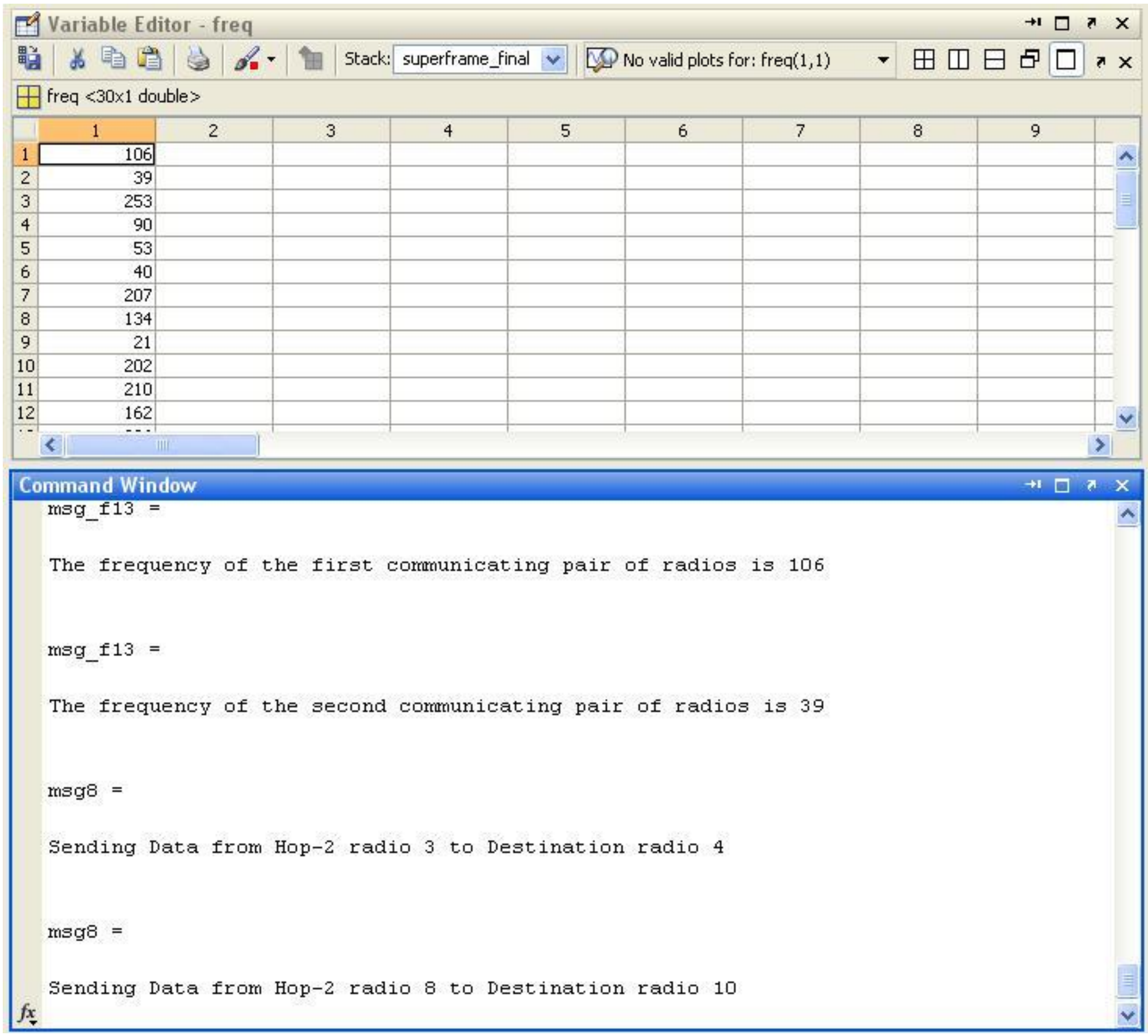


Figure 4.14 Frequencies allocated to the two subnets and transfer of data from hop-2 radio to the destination radio.

In the above figure 4.14 we can see that the frequencies in the first two frequency slots are allocated to the two subnets and the data is transferred from the hop-2 radios of the subnets to the destination radios of the subnets. In the next 10 time slots any data that the destination radio wants to transfer to the hop-2 radio is transferred in those 10 time slots in the specific time slot of the destination radio to the hop-2 radio. Also the two subnets start hoping on another frequency pattern. This can be seen below in figure 4.15,

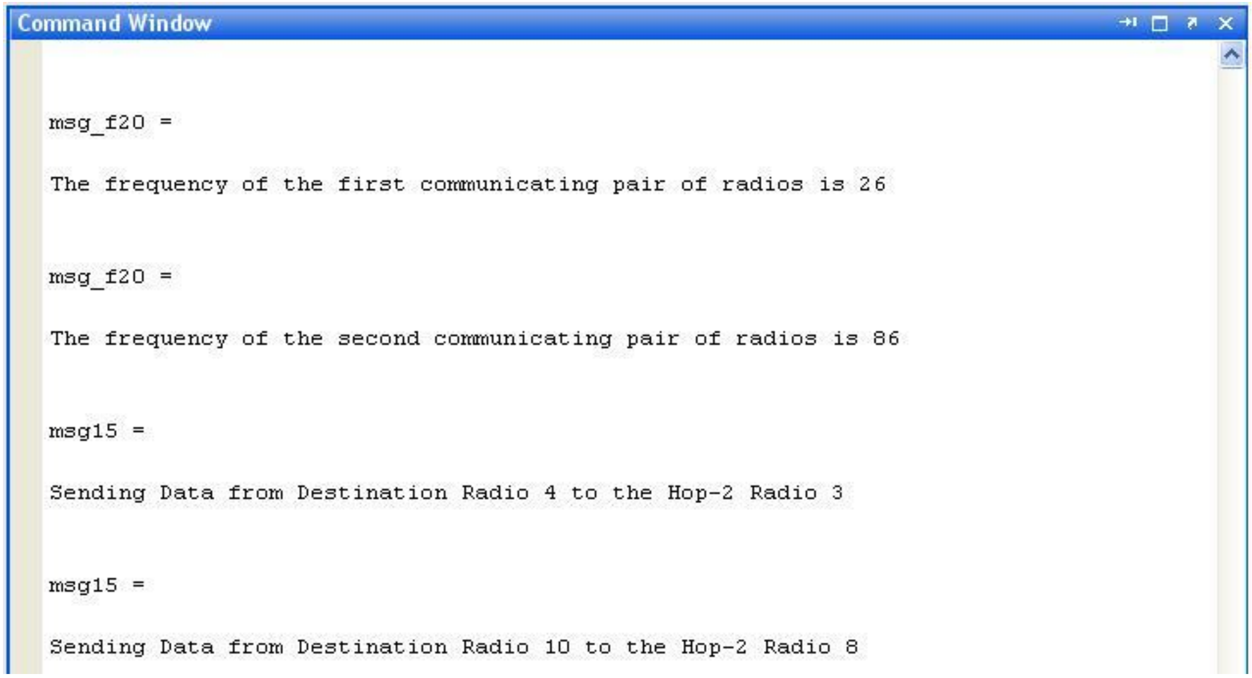


Figure 4.15 Frequencies allocated to the two subnets and transfer of data from destination radio to the hop-2 radio.

In the above figure we can see the data transfer from the destination radio to the hop-2 radio and the frequencies allocated to the two subnets. After that in the next 10 time slots the frequencies of the two subnets hop to another frequency and the data is transferred from the hop-2 radio to the hop-1 radio in the time slot of the hop-2 radio. This can be seen below in figure 4.16,

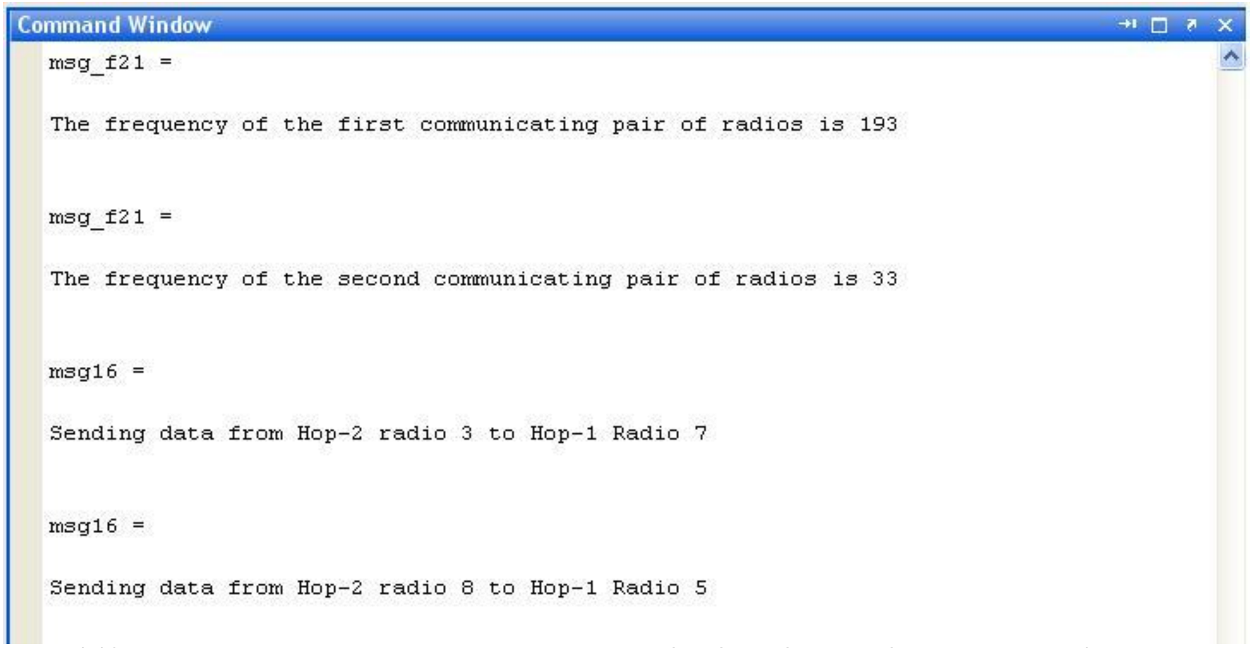


Figure 4.16 Frequencies allocated to the two subnets and transfer of data from hop-2 radio to the hop-1 radio.

In the above figure we can see that the frequencies of the two subnets have hoped to another frequency and that the data is being transferred from hop-2 radio to the hop-1 radio. In the next 10 time slots the frequencies of the two subnets hops to another frequency and the data is transferred from the hop-1 radio to the source radio. This can be seen below in figure 4.17,

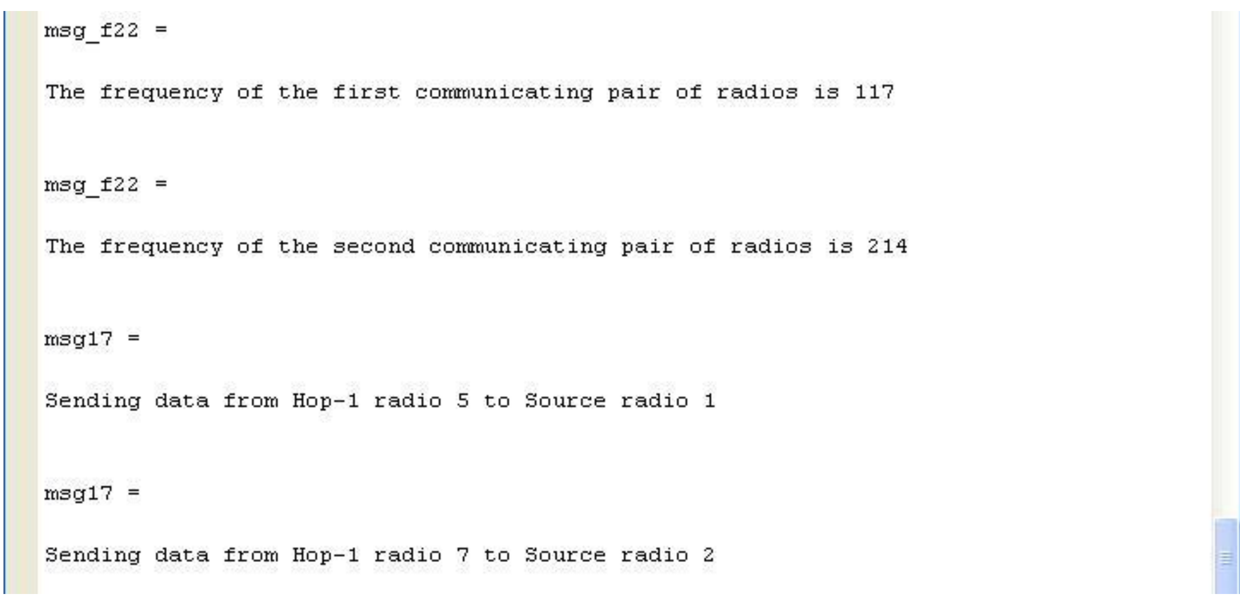


Figure 4.17 Frequencies allocated to the two subnets and transfer of data from hop-1 radio to the source radio.

From the above figure we can see that frequencies of the two subnets have hoped to another frequency and the data is being transferred from the hop-1 radio to the source radio in the time slot of the hop-1 radio. The two way communication from the source radio to the destination radio and then from the destination radio to the source radio, through staggered hopping in a multi-hop multi-subnet environment is completed in this MATLAB simulation.

4.3 Verilog Implementation

In our Verilog implementation we designed the proposed technique in Verilog. This was done after the MATLAB simulations were tested and complete. The source radio that was used was radio 1, the hop-1 radio that was used was radio 5 and the hop-2 radio that was used was radio 8 and the destination radio that was used was radio 10 for the subnet. All of the radios are using the same staggered frequency hopping for multiple access technique. As in the MATLAB simulations for the initial synchronization part of the super frame all of the radios in the subnet are hopping on the same frequency hopping pattern. As we are getting an 8 bit hashed output from the LFSR (Linear Feedback Shift Register) so the frequency is hopping among a total 256 frequency slots as $2^8=256$. These numbers that are generated in parallel are actually working as indexes for a RAM that would be used to select a frequency from the 256 pre-stored frequencies in the RAM to be forwarded to the transmitter. As our proposed technique is implemented on the Datalink layer (Layer 2) of the ISO layer stack the requests for hop-1, hop-2 and destination radio will be coming from the network layer. Figure 4.18 below shows the results of the implementation for the initial 30 time slots.

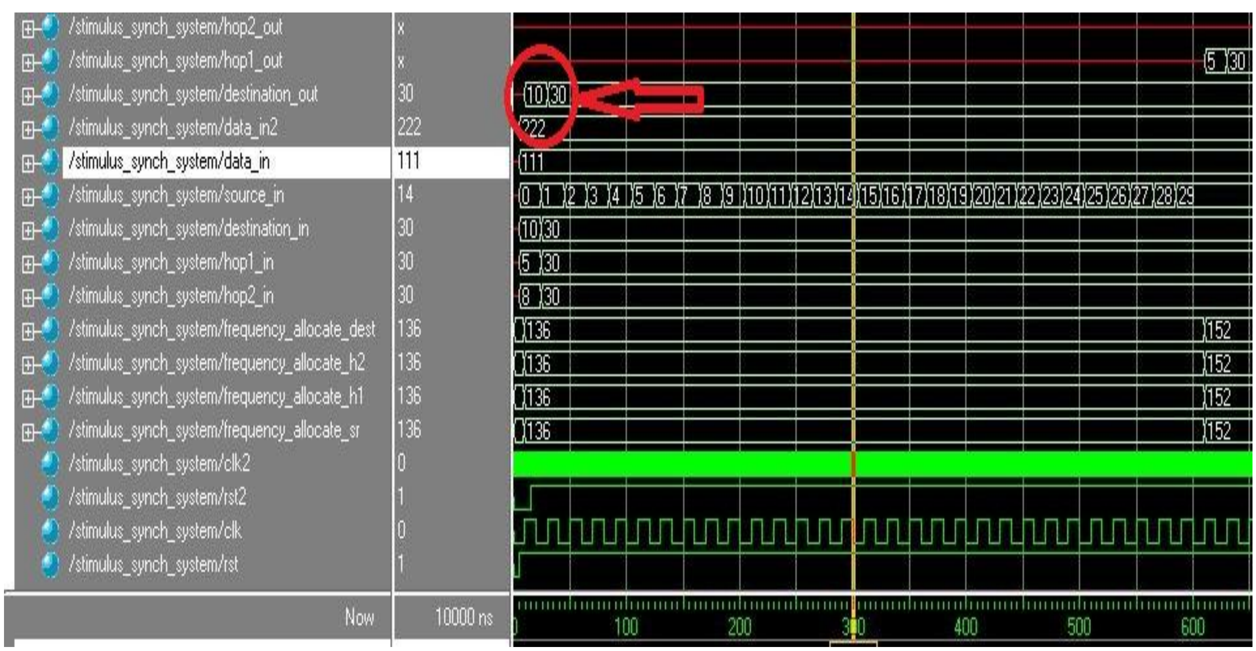


Figure 4.18 Destination radio requests are transmitted by the Source radio during the initial 30 time slots of the super frame.

From the above figure 4.18 we can see that during the initial 30 time slots the frequencies allocated to the source radio, hop-1 radio, hop-2 radio and destination radio are the same because the allocated frequency slot is 136.The signals used to represent the frequency allocated to the source radio is frequency_allocate_sr, hop-1 radio is frequency_allocate_h1, hop-2 radio is frequency_allocate_h2 and destination radio is frequency_allocate_dest. The source_in, hop1_in, hop2_in and destination_in signals are coming from the network layer (layer 3) to the datalink layer (layer 2) of the ISO layer stack. As we can see that signal for destination radio is radio 10, hop-1 radio is radio 5 and hop-2 radio is radio 8. These are passed down to layer 2 during the initial 30 time slots and also during the initial 30 time slots the source radio transmits the request for the destination radio. As we can see that in figure 4.18 during the first time slot the source radio transmits its request for destination radio by transmitting the number of the destination radio in the initial 30 time slots for destination radio request in its own specified time slot. As in our implementation we are only using a total of 30 radios with number from 0 to 29 and if a radio does not want to request a radio as a destination radio it transmits the number 30 instead. Each radio updates its table respectively as to which radio requested which radio to work as the destination radio. So therefore we can see from the figure 4.18 that during the first time slot the source radio transmits a request for radio 10 which is the destination radio this is denoted by the red encircled time slot and a red arrow pointing towards it and after the first time slot in which the source radio transmits its request for the destination radio the other do not transmit request for destination radio and transmit the number 30 instead. After the first 30 time slots the frequency of all of the radios hops to another frequency which is in frequency slot 152 and the source radio then transmits its request for hop-1 radio which is radio 5 which can be seen from figure 4.19. In the following time slots after the first time slots which is for the first radio which is working as the source radio, the next 29 time slots each radio transmits 30 as they are not requesting any radio to work as hop-1 radio for them. The source radio which is radio 0 request is show by the circle which is being pointed to by the red arrow in the figure 4.19.

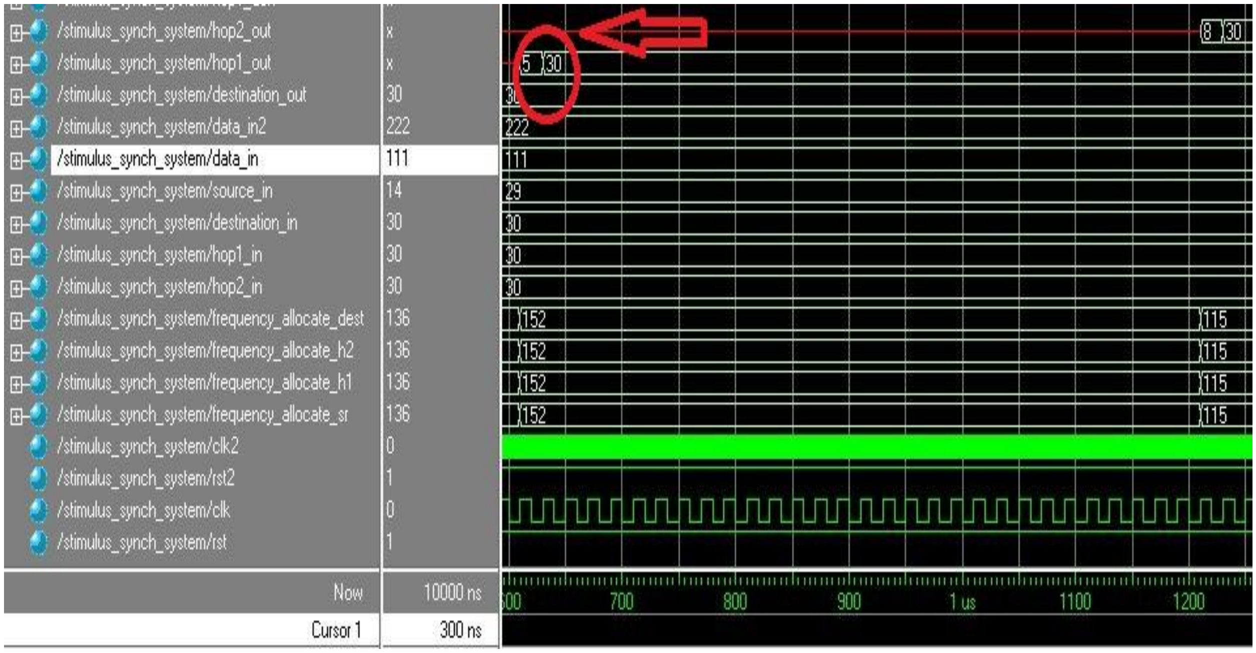


Figure 4.19 Hop-1 radio requests are transmitted by the Source radio during the next 30 time slots of the super frame.

Similarly all of the other radios update their tables respectively as to which radio requested which radio to work as hop-1 radio. In the next 30 time slots the frequency hops to the frequency in the frequency slot 115 and the source radios transmit there request for hop-2 radio which is radio 8 and is denoted by the red encircled time slot which is being pointed to by the red arrow. This is shown below in the figure 4.20,

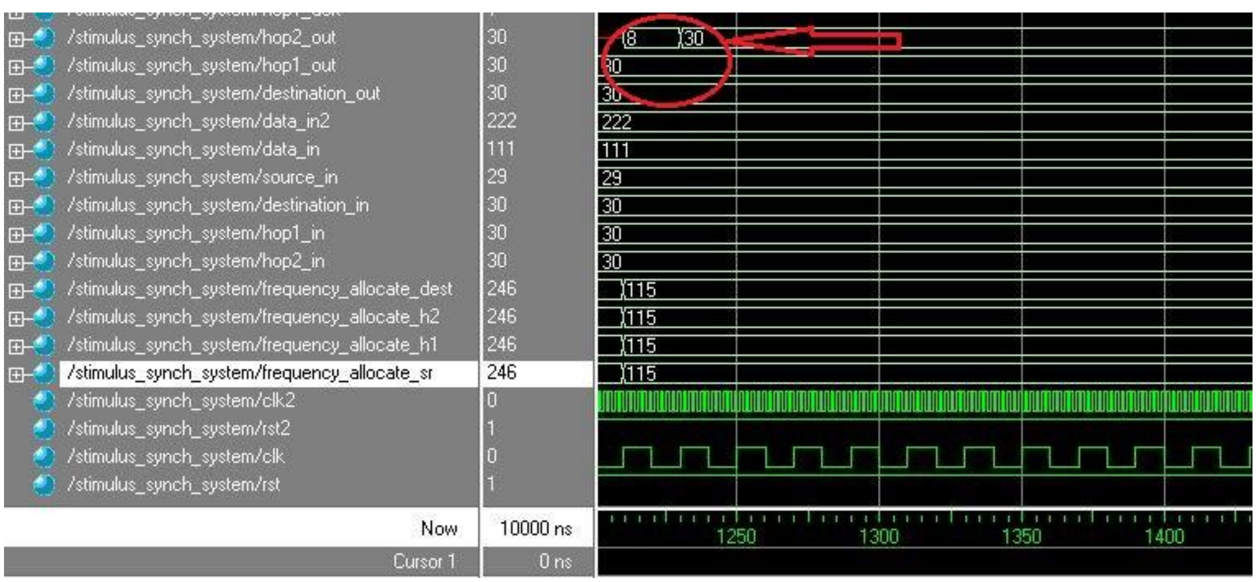


Figure 4.20 Requests for Hop-2 Radio by the Source Radios.

As we can see from figure 4.20 during these 30 time slots the source radio which is radio 0 transmits its request for the hop-2 radio which is radio 8 during its specified time slot. As no other source radio is transmitting a request for hop-2 radio all of the other radios transmit 30 in their own specified time slots. The signals used to represent the hop-2 request is given by hop2_out. After these 30 time slots in the next 30 time slots the frequency hops to another frequency which is the frequency in the 194th frequency slot and the radios that are working as hop-1 radio transmits an acknowledgement to the source radio if it wants to work as a hop-1 radio for the source radio.

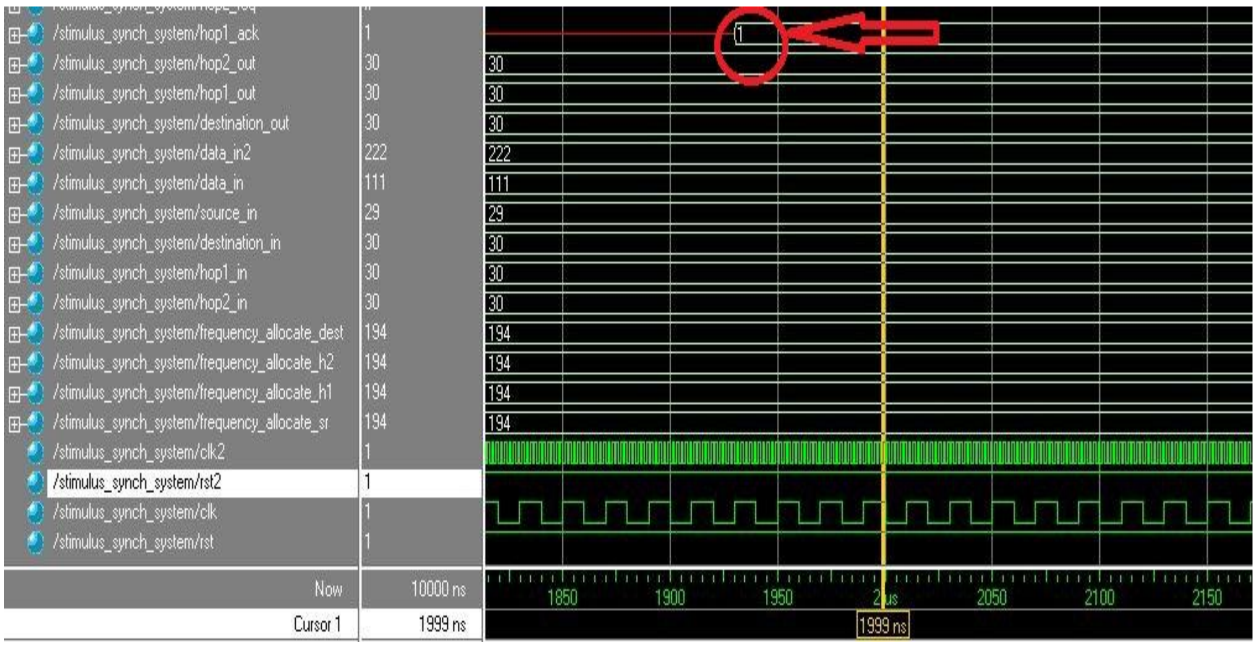


Figure 4.21 Hop-1 Radio Transmits an Acknowledgement.

As we can see from the above figure 4.21 that the radio 5 which was requested to work as hop-1 radio by the source radio 0, transmits its acknowledgement in its own specified time slot which is the fifth time slot. The hop-1 radio transmits 1 as an acknowledge and if it does not want to work as a hop-1 radio or has not been requested to work as a hop-1 radio for a source radio then it transmits a zero. From the above figure we can see that the hop-1 radio 5 transmits a 1 in its own time slot which is represented by the red encircled time slot which is being pointed to by the red arrow in figure 4.21. The signal used to represent the acknowledgement from hop-1 radio is hop1_ack. All of the other radios update their tables respectively as to which radio transmitted an acknowledgment to work as a hop-1 radio for a source radio. In the next 30 time slots frequency hops to another frequency which is the frequency in the 133th frequency slot and the hop-1 radio transmits a request for hop-2 radio by transmitting the number of the hop-2 radio in its own specified time slot that was requested by the source radio to work as hop-2 radio for it alongside the hop-1 radio. This is shown in figure 4.22 below,

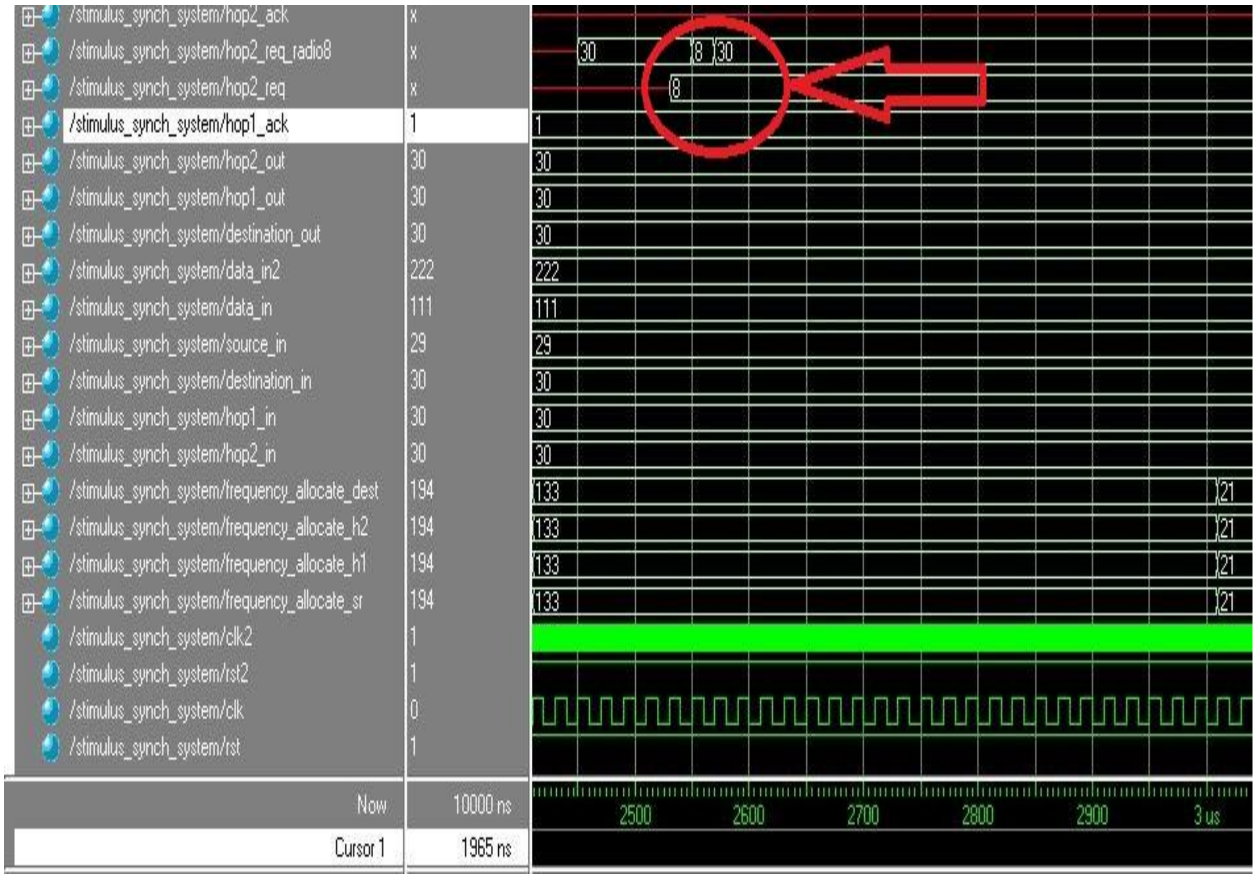


Figure 4.22 Hop-1 radio requests for hop-2 radio and hop-2 radio acknowledgment.

As we can see from the above figure 4.22 that the hop-1 radio transmits its request for hop-2 radio in its own time slots that is the fifth time slot. The signal hop2_req is used to represent the hop-1 radio request for hop-2 radio. The signal hop2_req_radio8 shows the request stored in the 5th entry for the array in the table of the hop-2 radio as the request is transmitted in the 5th time slot of the 30 time slots for hop-1 radios to request hop-2 radios both these signals are shown by the red encircled time slot and the red arrow pointing towards it. All of the radios update their respective tables as to which radio requested which radio to work as hop-2 from hop-1 radio for source radio. During the next 30 time slots the radios that were requested to work as hop-2 radios from hop-1 radio transmit an acknowledgement to the hop-1 radio and the frequency hops to another frequency which is the frequency in the 21st frequency slot. This can be seen in the figure 4.23 in the 30 time slots after the 30 time slots for hop-1 radio to request hop-2 radios. The hop-2 radio which is radio 8 transmits an acknowledge in its own time slots to the hop-1 radio which is represented by the red encircled time slot which is being pointed to by the red arrow in figure 4.23, all of the radios do receive this acknowledge and update their tables as to which radio transmitted an acknowledgment to work as hop-2 radio.

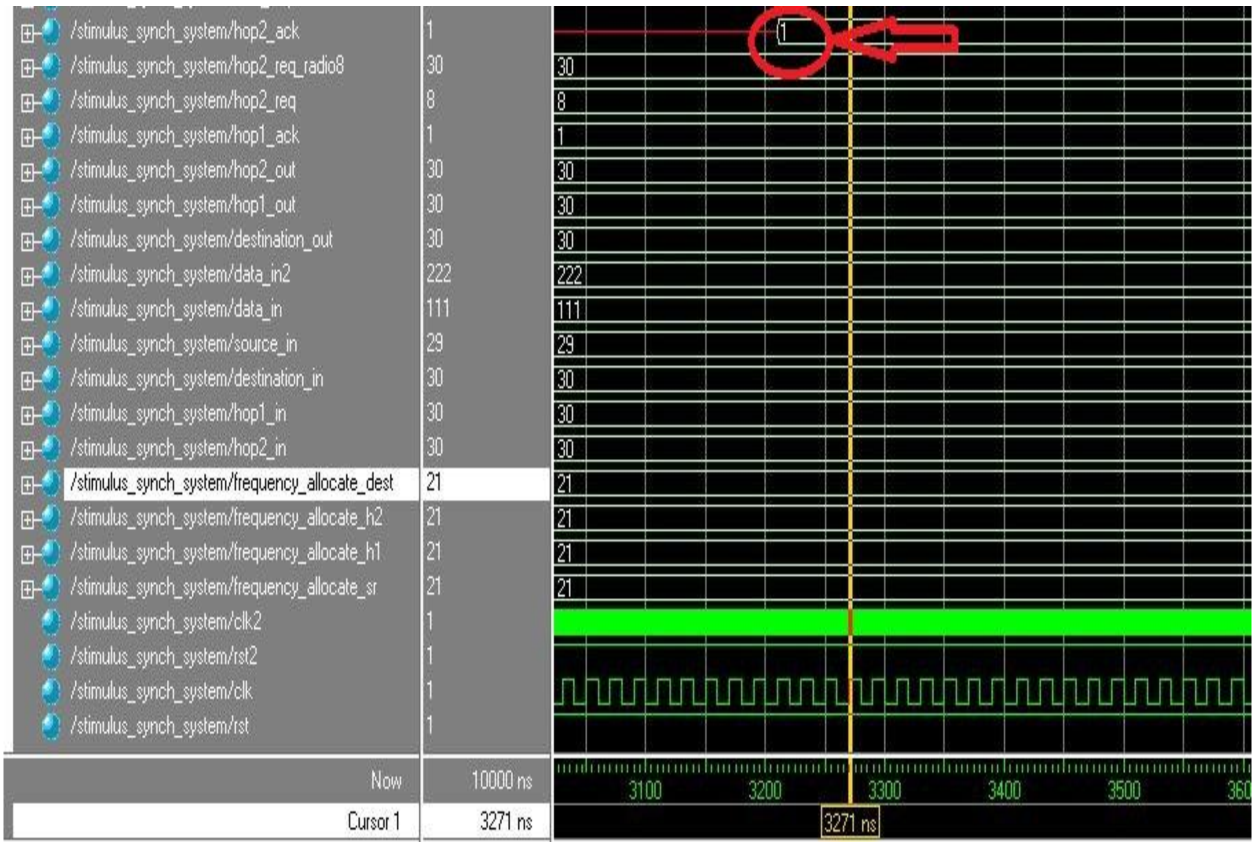


Figure 4.23 Hop-2 Radio transmits an Acknowledgement

In the next 30 time slots the frequency hops to another frequency which is the frequency in the 11th frequency slot, the hop-2 radio transmits its request for destination radio which is shown below in figure 4.24,

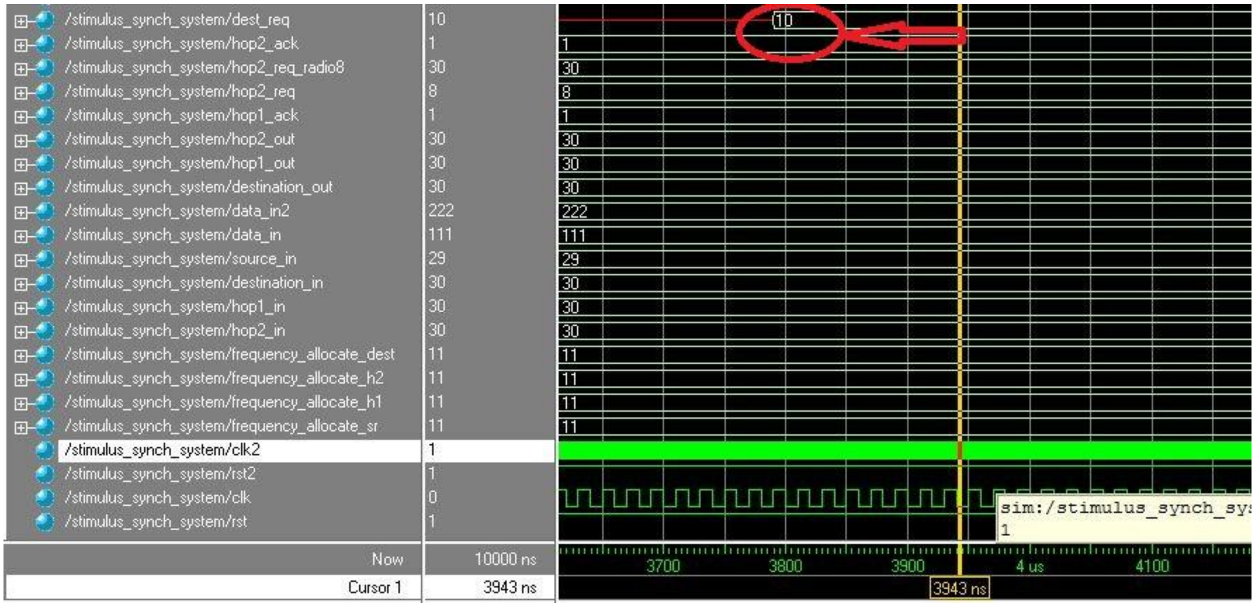


Figure 4.24 Hop-2 radio transmits requests for the destination radio.

In the above figure 4.24 we can see that the hop-2 radio which is radio 8 transmits its request for the destination radio in its own specified time slot that is the 8th time slot which is denoted in the figure 4.24 by the red encircled time slot. All of the other radios update their tables respectively as to which hop-2 radio requested which destination radio for which source radio. In the next 30 time slots the frequency on which all the radios are hopping hops from the frequency in the 11th frequency slots to the frequency in the 54th frequency slot and the destination radio transmits an acknowledgment in its own time slot which is again denoted in the figure 4.25 below by the red encircled time slot.

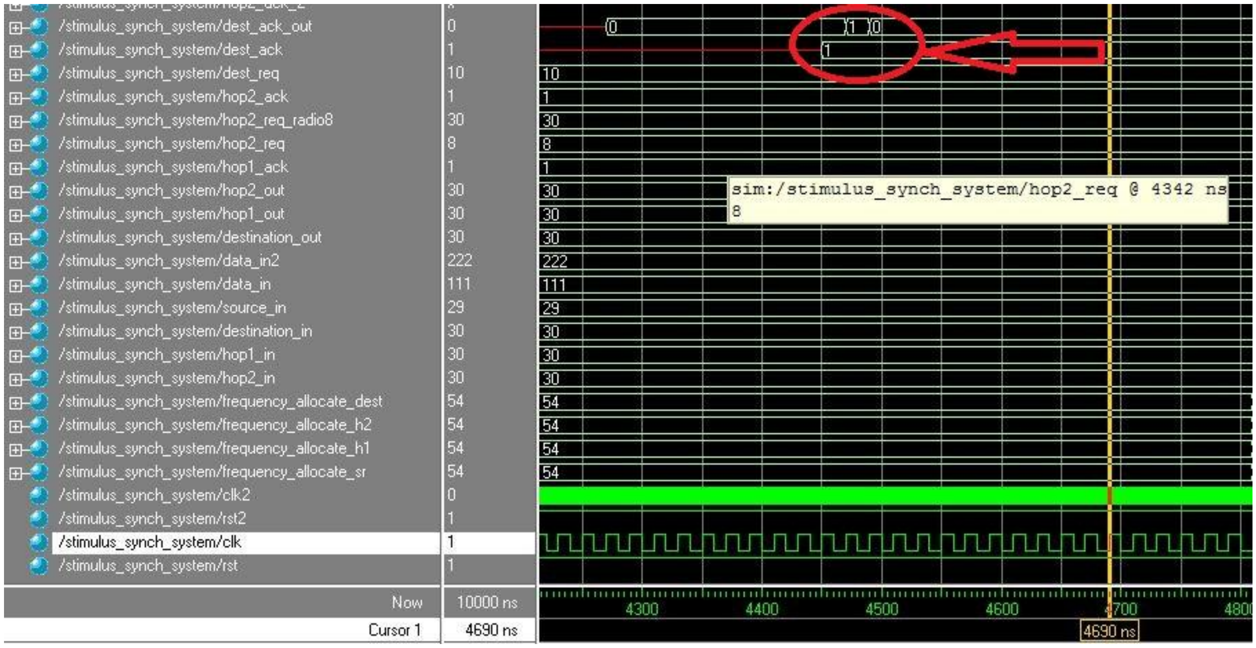


Figure 4.25 Destination Radio transmits an Acknowledgement

The signal dest_ack is used to represent the destination acknowledge signal. As we can see that in figure 4.25 the destination radio which is radio 10 transmits an acknowledgment in the tenth time slot and since no other radio has been requested to work as a destination radio the other radios transmit a zero in their own specified time slot. The signal dest_ack_out show the destination acknowledgment has been stored in the array in the table of the hop-2 radio; all of the other radios similarly update their tables as to which radio working as a destination radio transmitted an acknowledgment to which radio working as a hop-2 radio.

In the next 30 time slots the frequency hops from the frequency in the 54th frequency slot to the frequency in the 23rd frequency slot and hop-2 radio transmits its second acknowledgement during its own time slot this can be seen from figure 4.26 below,

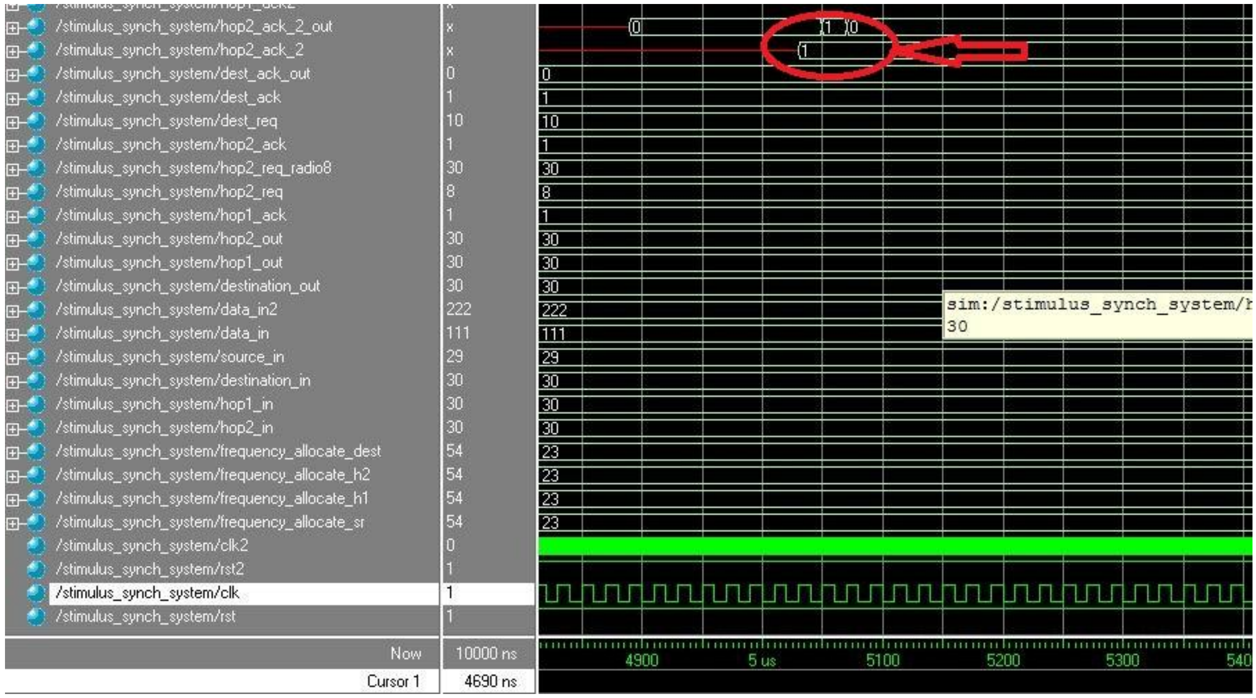


Figure 4.26 Hop-2 radio transmits its second acknowledgment.

From the above figure 4.26 we can see that the hop-2 radio transmits its second acknowledgment during its own timeslots that is the 8th time slot this is shown by the signal hop2_ack_2 and the signal hop2_ack_2_out shows the acknowledgment is stored in the array of the table of the hop-1 radio which is denoted by the time slots encircled in red and the red arrow pointing towards the encircled timeslots. After this in the next 30 time slots the frequency hops again to another frequency in the 152nd frequency slot. The hop-1 radio transmits its second acknowledgment to the source radio in its own specific time slot all the other radio update their tables respectively during each time slot this is shown below in the figure 4.27.

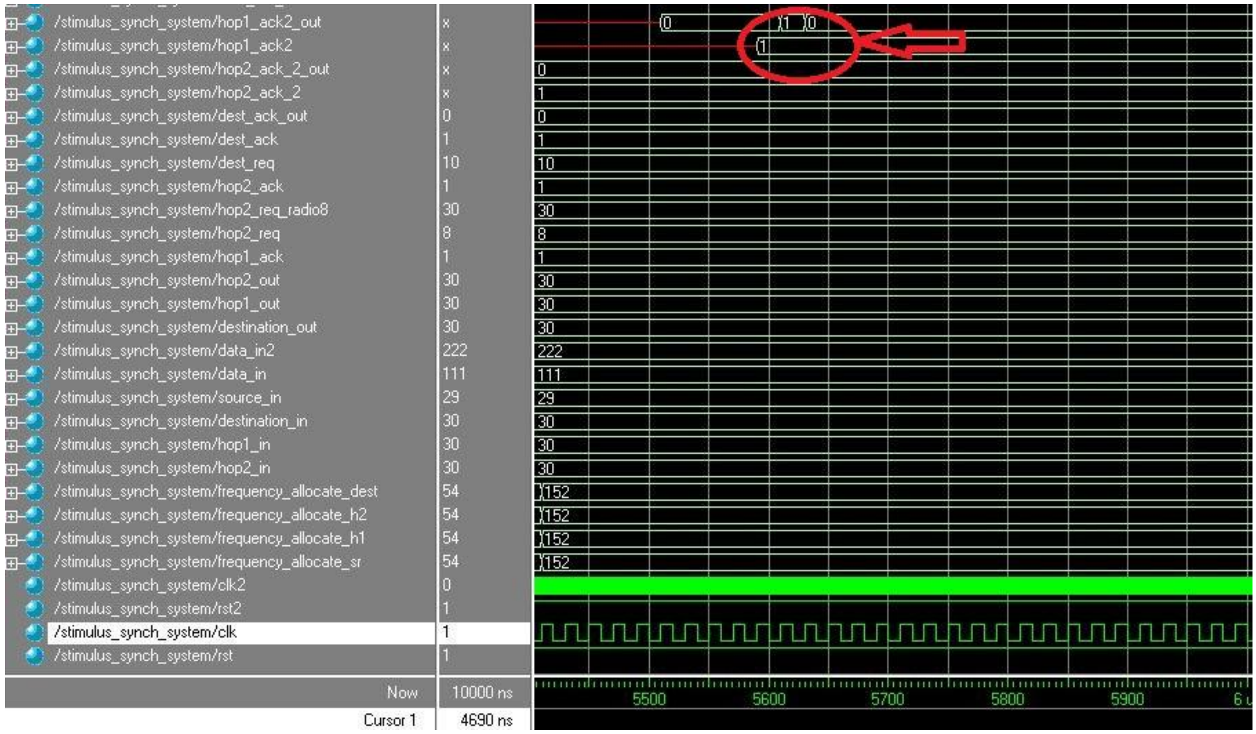


Figure 4.27 Hop 1 Radio transmits its second Acknowledgement.

In the above figure 4.27 the signal hop1_ack2 represents the hop-1 radio second acknowledgment signal and the hop1_ack2_out show that hop-1 radio acknowledgment stored in the source radio these are represented by the red encircled time slots.

Once this initial creation of subnets part of the super frame is complete then each subnet starts to hop on its own frequency hopping pattern. The frequency hopping pattern of the subnet is the hopping pattern that is in the frequency slot number of the number of the source radio. That is if the source radio of a subnet is radio 5 then the frequency hopping pattern in the fifth frequency slot will be used for that particular subnet. It should be noted that however the radio still will transmit the data in their own specific time slots out of the 30 time slots. In the next 30 time slots each subnet starts hopping on its own frequency hopping pattern as the source radio is radio 0 for the subnet in our simulation the radios in this subnet will start to hop on the frequency in the frequency slot 0 and the source radios of each subnet will start transferring their data the source radio 0 will start transferring the data 111 to the hop-1 radio as these 4 radios are hopping on their own frequency hopping pattern. This can be seen below in figure 4.28,

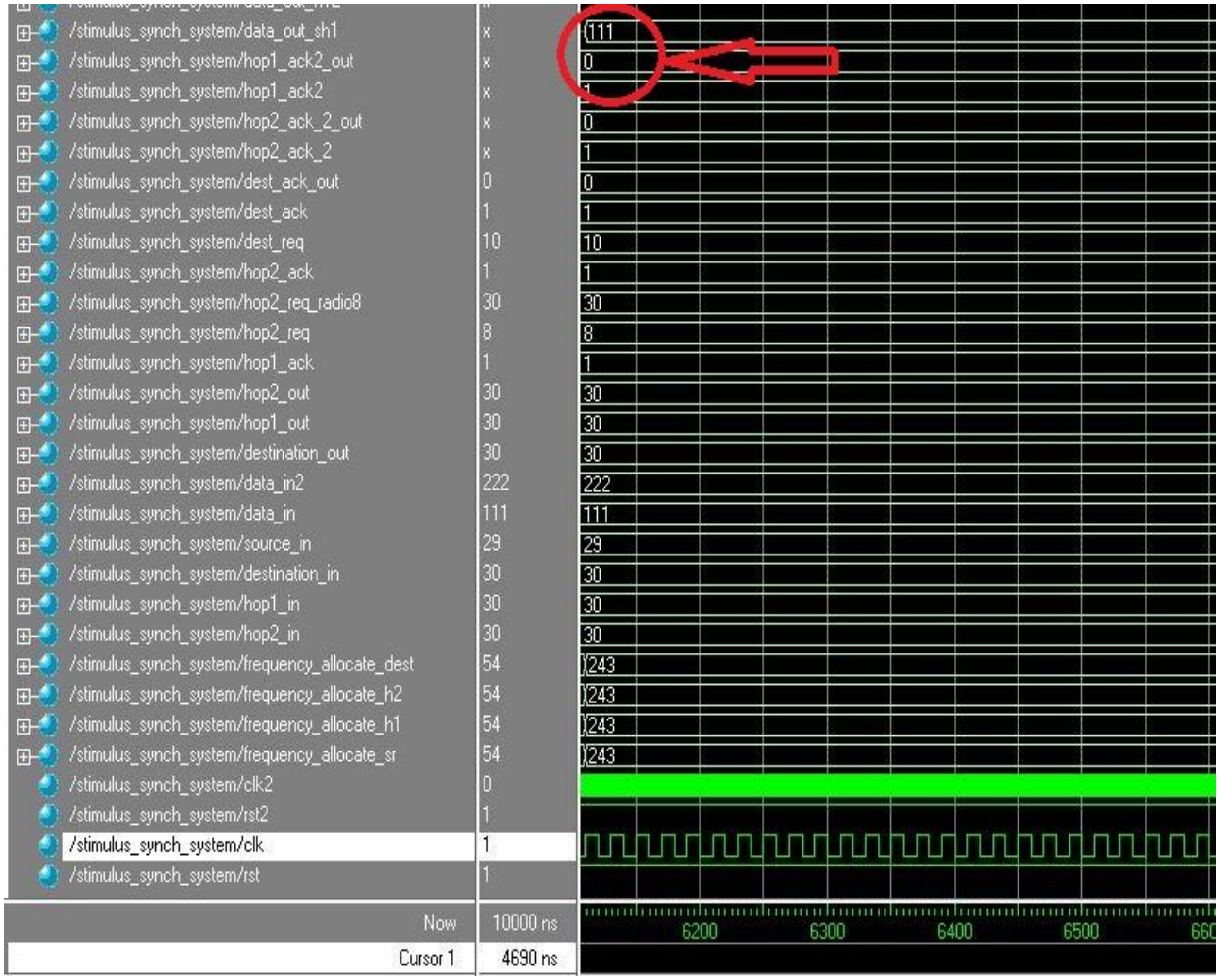


Figure 4.28 Source Radio Transmits Data to Hop-1 radio.

From the figure 4.28 above, we can see that the source radio transmits its data to the hop-1 radio and the time slots are encircled in red and pointed to by a red arrow. In the following 30 time slots the hop-1 radio transmits data to the hop-2 radio this is shown in figure 4.29 below by the time slots encircled in red with the arrow pointing towards them,

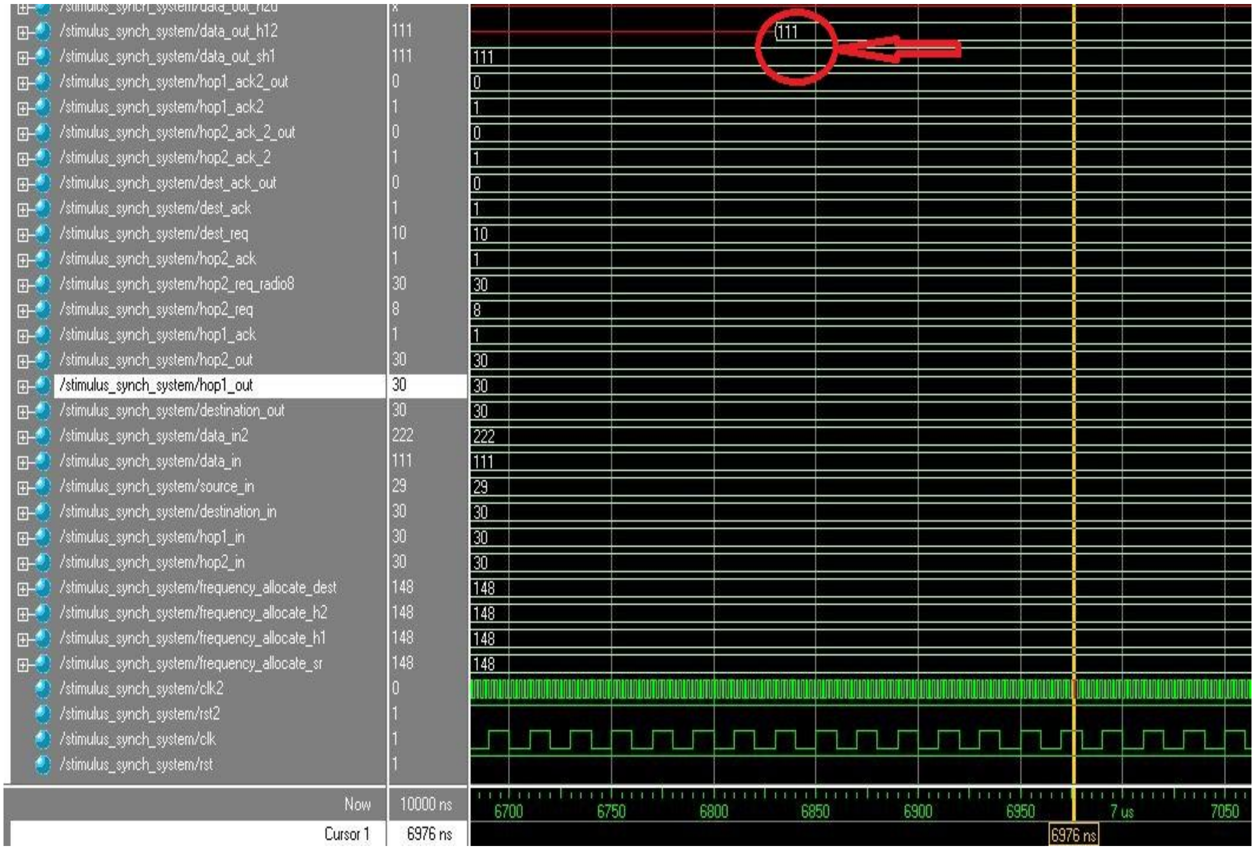


Figure 4.29 Data transfer from Hop-1 Radio to Hop-2 Radio

The radios in a particular subnet are communicating on a frequency that is hoping after every 30 time slots. In figure 4.30 we can see that then hop-2 radio transmitting data from the hop-2 radio to the destination radio this is also denoted by the red encircled time slots.

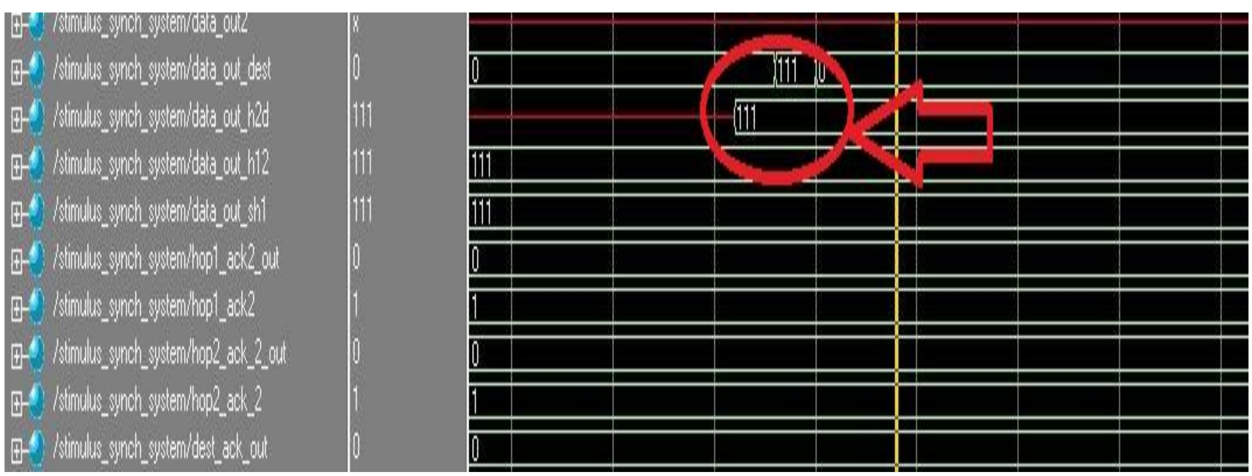


Figure 4-30 Data Transfer from Hop-2 Radio to the Destination Radio.

In the figure 4.30 above the signals data_out_h2d represents the data that is being transmitted from the hop-2 radio to the destination radio. The signal data_out_dest shows the data that is stored in the destination radio these are both encircled in red and have a red arrow pointing at them. After this in the next 30 time slots the data ‘222’ is transmitted from the destination radio to the hop-2 radio in the time slot of the destination radio. This is shown in the figure 4.31 below,

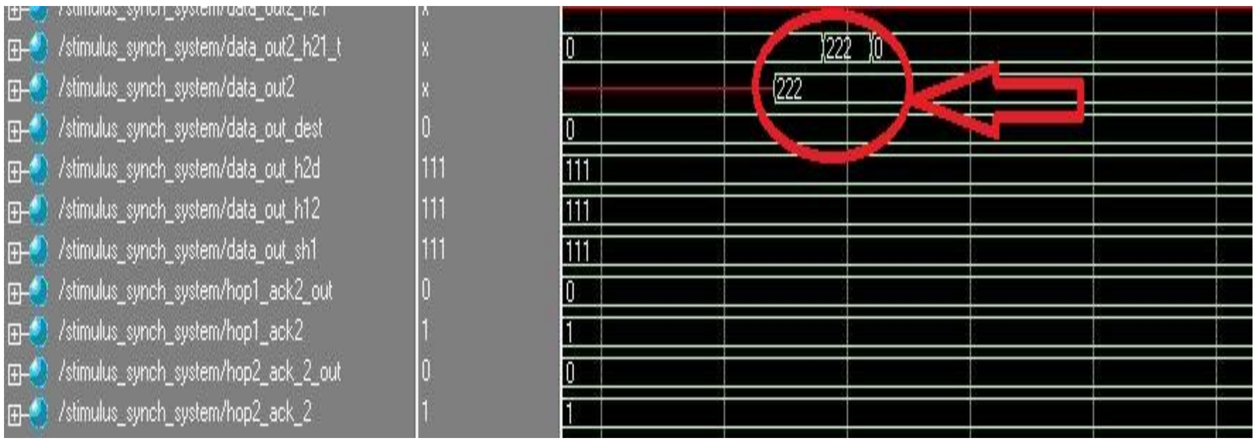


Figure 4.31 Data transmitted from Destination radio to the Hop-2 radio.

In the above figure 4.31 we can see that the destination radio transmits data to the hop-2 radio. The signal data_out2 shows the data transmitted from the destination radio to the hop-2 radio and the signal data_out2_h21_t shows the data stored in the hop-2 radio both of the time slots are encircled in red and have a red arrow pointing at them. After this in the next 30 time slots the hop-2 radio transmits data to the hop-1 radio in its own specified time slot. This is shown below in figure 4.32,

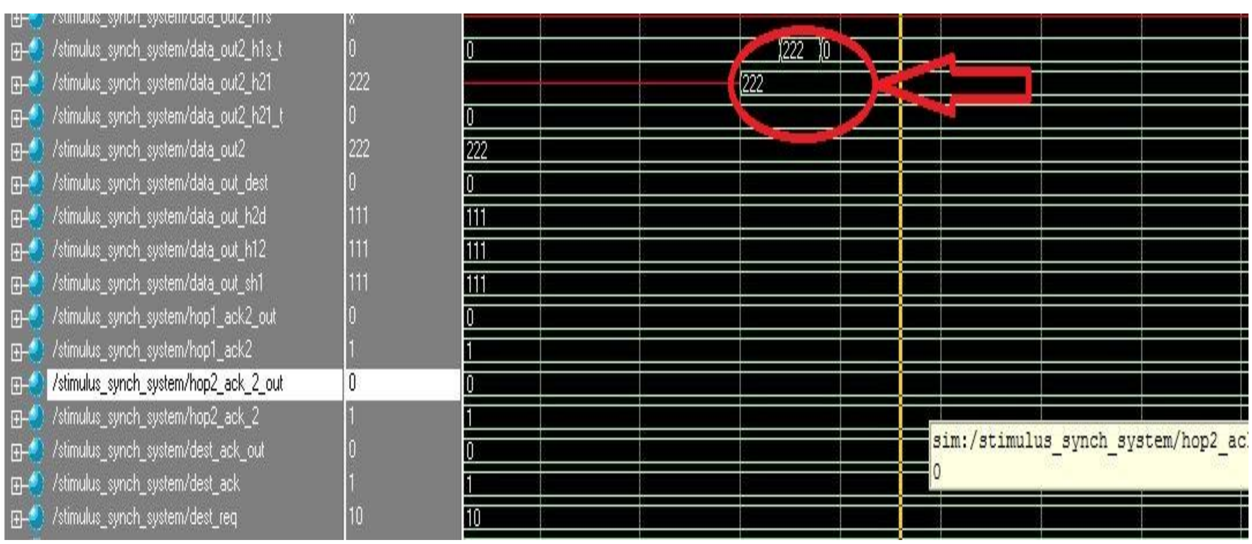


Figure 4.32 Data transfer from Hop-2 Radio to the Hop-1 Radio

In the above figure 4.32 we can see that the hop-2 radio transmits data from itself to the hop-1 radio in its own time slots out of the 30 timeslots these are denoted by the red encircled time slots and the arrow pointing towards them. In the figure 4.32 the signal data_out2_h21 shows the data being transmitted from the hop-2 radio to the hop-1 radio and the signal data_out2_h1s_t shows the data stored in the hop-1 radios array in its tables. All the other radios also update their tables as to which hop-2 radio transmits data from itself to the hop-1 radio. In the next 30 time slots the hop-1 radio transmits the data from itself to the source radio. This can be seen from the figure 4.33,

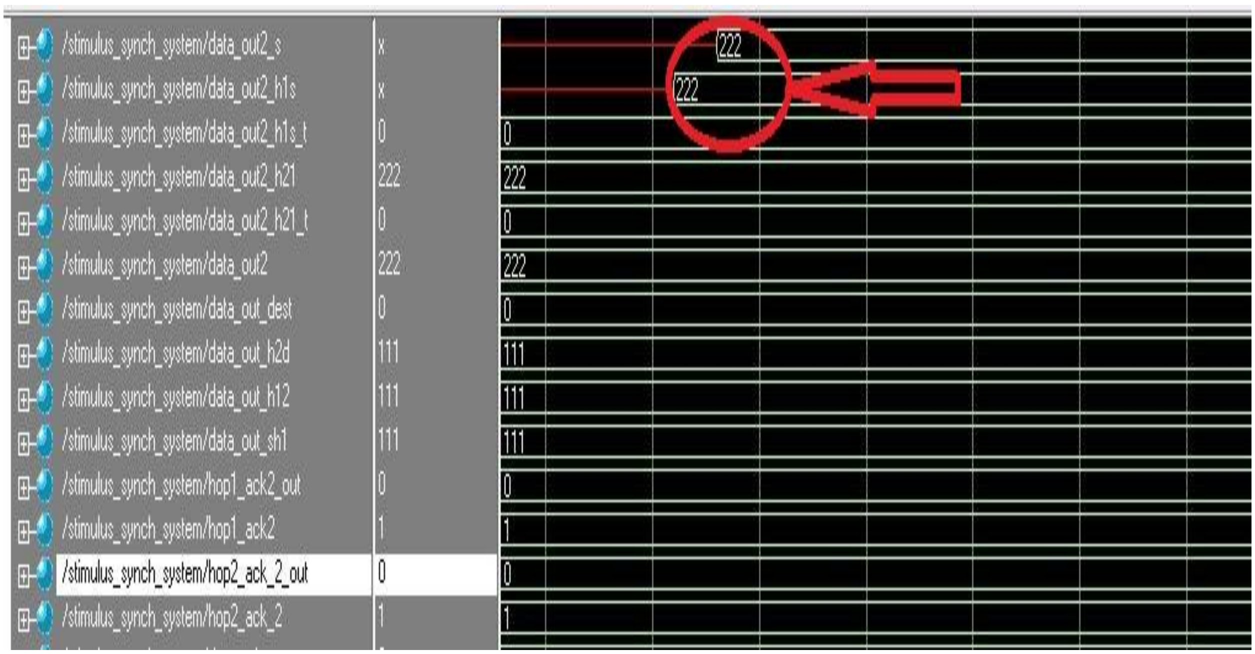


Figure 4.33 Data transfer from the Hop-1 radio to the Source Radio

From figure 4.33 we can see that the hop-1 radio transmits the data to the source radio in its own time slots which is represented by the red encircled time lot and the arrow pointing towards it. The signal data_out2_h1s represents the hop-1 radio transmitting data to the source radio in its own time slot. Also the signal data_out2_s represents the data that is stored in the source radio after the data was transmitted by hop-1 radio and stored in the source radio. In the next 30 time slots the super frame starts again and all of the radios start hoping on the same frequency again.

Summary:

In this chapter the implementation and results are shown for the proposed technique in both MATLAB and Verilog. Also the analysis from the results are concluded which shows successful staggered hopping in a multi-subnet multi-hopped environment.

Conclusion and Future Work

Chapter 5

5.1 Conclusion

In recent years there has been much work and research done in both the fields of multi-hoping and frequency hopping as the applications of wireless solutions grows in the field of multi-hoping. Research has been conducted in the fields of multi-hoping as to how increasing or decreasing the number of hops affected power and comparisons of different routing algorithms in a multi-hoped network. In our thesis we have focused on a new technique that would provide better security for multi-hop networks by using staggered hopping in a multi-subnet multi-hop environment.

In this thesis, we implemented a new technique for frequency hopping in a staggered multi-hop multi-subnet environment supporting up to two hops and n radios. This technique was implemented on the ISO Datalink Layer (Layer 2) and successfully synchronized and implemented the communications between radios in different subnets while frequency hopping in a staggered multi-hop multi-subnets environment. In our simulation we have successfully implemented and tested this technique. In our thesis we are using multi-hoping and creating different subnets for groups of radios to communicate in for increased security as different subnets that are hopping on different staggered frequency hopping patterns while communicating data are more immune to eavesdropping and other vulnerabilities. Initially the algorithm was implemented in MATLAB and then was designed in Verilog using ModelSIM. The proposed technique was successfully implemented for a total of 30 radios and subnets supporting groups of 4 radios in which there is a source radio, hop-1 radio, hop-2 radio and destination radio.

5.2 Future Work

Future work is aimed to increase the number of hops between the source and destination radio and make the algorithms adaptive to work for a variable number of hops between the source and destination radio. Also the number of radios in a group of subnet could be increased and also made adaptive alongside a performance analysis.

References

[1] J. Min and H. Samueli, “Synchronization Techniques for a frequency hopped wireless Transceiver”, *Journal*, Publisher, Irvine, CA, 28 Apr-1 May 1996, pp. 183-187.

[2] Pearson, Peter K.,” Fast Hashing of Variable-Length Text Strings”, Communications of the ACM 33, June 1990 (6): 677.

[3] Per Beming and Magnus Frodigh, “Admission Control in Frequency Hopping GSM systems”, Vehicular Technology Conference, 1997, IEEE 47th, 1282 – 1286.

[4] Rongxin Zhi, Luyong Zhang, Zheng Zhou, ”Cognitive Frequency Hopping”, 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008.

[5] Jonathan S. Min and Henry Samueli, “Analysis and Design of a Frequency-Hopped Spread-Spectrum Transceiver for Wireless Personal Communications”, IEEE Transactions on Vehicular Technology, vol. 49, no. 5, September 2000. Page(s): 1719 – 1731.

[6] Teemu Vanninen, Harri Saarnisaari, Matti Raustia and Timo Koskela, “FH-Code Phase Synchronization in a Wireless Multi-Hop FH/DSSS Adhoc Network”, MILCOM 2006,pp.1-7.

[7] Qi Ling, Tongtong Li, Zhi Ding, “A Novel Concept: Message Driven Frequency Hopping (MDFH)”, Communications, 2007. ICC '07. IEEE International Conference, 24-28 June 2007

[8] Zoran Kostic and Nelson Sollenberger,"Dynamic Frequency Hopping for Limited-Bandwidth Cellular Systems", Computer Communications and Networks, 2000. Proceedings, Ninth International Conference, 06 August 2002, pp.28 - 33

[9] Wan Yadong, Wang Qin, Duan Shihong, Zhang Xiaotong, "RAFH: Reliable Aware Frequency Hopping Method for Industrial Wireless Sensor Networks", Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference, 24-26 Sept. 2009

[10] M. Abolhasan, J. C.-P. Wang, D. R. Franklin, "On Indoor Multi-hopping capacity of Wireless Ad-Hoc Mesh Networks", IEEE International Conference Mobile Ad-Hoc and Sensor Systems, 2007.

[11] Pedro H. J. Nardelli and Giuseppe T. F. de Abreu, "Analysis of hopping strategies in multi-hop wireless networks", 6th Workshop on Positioning, Navigation and Communication, 2009. WPNC 2009. pp.117 - 123

[12] Fujiwara, T., Iida, N., Watanabe, T.,"A Wireless Network Scheme Enhanced with Ad-Hoc Networking For Emergency Communications", IEEE International Conference on Communications, 2004, pp. 4177 - 4181 Vol.7.

[13] Bjornemo, E., Johansson, M., Ahlen, A., "Two Hops Is One Too Many In An Energy-Limited Wireless Sensor Network", IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007, pp. III-181 - III-184.

[14] Yuh-Shyan Chen and Chao-Yu Chiang, "PER A Power-Life Extension Routing Protocol Using a Round Robin Scheme for Mobile Ad Hoc Networks", Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, 2004. Page(s): 5 - 8 Vol.1.

[15] Sharony, J., "A mobile radio network architecture with dynamically changing topology using virtual subnets", IEEE International Conference on Communications Converging Technologies for Tomorrow's Applications. 1996, Page(s): 807 - 812 vol.2.

[16] Weihuang Fu, Xiaoyuan Wang, and Dharma P. Agrawal, "Multi-subnets Selection and Rate Allocation in a Heterogeneous Wireless Network", IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th, Page(s): 715 – 720.

[17] Pearson, Peter K.," Fast Hashing of Variable-Length Text Strings", Communications of the ACM 33, June 1990 (6), Page(s): 677.

[18] http://www2.ece.ohio-state.edu/~roblin/freqhop/f2_2a.xbm

[19] http://en.wikipedia.org/wiki/Time_division_multiple_access

[20] www.complextoreal.com/chapters/mod1.pdf