Implementation of Interface Agility for Duplex Dynamic Spectrum Access Radio Using USRP

Suman Bhunia, Shamik Sengupta
Dept. of Computer Science and Engineering, University of Nevada, Reno, USA
sbhunia@nevada.unr.edu, ssengupta@unr.eduu

***Abstract*—Dynamic Spectrum Access (DSA) based Cognitive Radio (CR) allows a secondary user (SU) to use spectrum in an opportunistic manner when the primary user (PU) of that spectrum is not transmitting. Conventional dynamic spectrum allocation methods deal with acquiring spectrum as fixed width channels. In this paper, we propose a novel technique where a node inside a network can dynamically access the spectrum of any width. We also show that a node can create multiple virtual interfaces to communicate with multiple neighbors simultaneously. With a prototype built with off-the-shelf software defined radio, we indicate that a node can achieve duplex communication with multiple neighbors utilizing a single transceiver.**

Keywords*—*Virtual interface, DSA, GNURadio, USRP

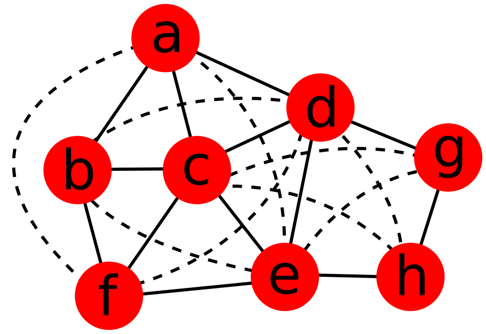


Fig. 1: Sample multihop network

I. Introduction

Over the last decade, wireless applications' demand has increased substantially. Research and innovation of sophisticated wireless devices have increased data rate to more than 100 times in past 20 years. The widespread availability of advanced physical layer communication technology such as orthogonal frequency division multiplexing (OFDM), multiple inputs multiple outputs (MIMO), etc. have escalated the procedure. In this line, the software defined radio (SDR) is the next technology that brings another freedom regarding frequency agility. Unlike conventional radios, SDR allows a system to change physical layer parameters (e.g. center frequency, bandwidth, power, modulation techniques, etc.) dynamically by its internal processing or decision-making unit.

However, the system becomes very complex and inefficient when heterogeneous radios [1] are present. Conventional multiple access methodologies include either frequency-division or time-division multiplexing. Most of the FDM techniques are inherently fixed bandwidth in nature. Wi-Fi system can change its frequency of operation but have to maintain its fixed bandwidth requirement. On the other hand, time division multiplexing can be a fixed time-based polling system or nodes can wait to access the spectrum in a contention window based system. Usually, nodes use contention window based system when there is no central controller to manage the channel access. This results in an exponential decrease in throughput with the increase in the number of contending nodes [2]. Figure 1 illustrates a sample multihop network. One hop neighbors who can communicate with each other are linked by solid lines, whereas the dotted lines represent the interference links. Here, nodes a and f can not communicate with each other due to distance, but they interfere with each other if they use the same spectrum. One solution to the interference problem is to use heterogeneous radios that operate on different frequencies. Conventionally, to connect heterogeneous radios, a device need to have multiple interfaces through which it can communicate with other devices. An alternative approach is to use SDR and connect it to each radio at various times while adapting suitable system parameters in that time interval. This method requires huge delay and proper time domain coordination with all the neighbors in a mesh network.

The problem of DSA has been a focus of research for several years [3]. However, investigating channel aggregation and fragmentation has recently gained some attraction. While traditional spectrum allocation algorithms assign contiguous channels to users, wireless techniques such as NC-OFDM [4] provide the possibility of spectrum aggregation, in which multiple spectrum holes can be joined to satisfy the bandwidth requirements of a user [5]. *Aggregation Aware Spectrum Assignment (AASA)* [6] is one of the earlier spectrum aggregation algorithms presented in the literature. This greedy algorithm is developed based on the assumption that all users require the same amount of spectrum and uses a first-fit approach for channel assignments. In contrast to AASA, Maximum Satisfactory Algorithm (MSA) [7] is a best-fit algorithm developed for the case where users may have different spectrum requirements. In this approach, users with higher bandwidth requirements are prioritized as they are more difficult to fit in a narrower spectrum holes. *Channel Characteristic Aware Spectrum Aggregation algorithm (CCASA)* [8] considers the heterogeneity of data carrying capacity in different parts of the spectrum. Once the channel state information of all users is known, a CCASA central controller allocates suitable spectrum fragments to the user by utilizing NC-OFDM. Using a sliding window method, CCASA calculates the maximum spectrum usage ratio for each user and allocates the spectrum to users in decreasing order of their spectrum requirements. The work presented in [9] investigates fragmentation and aggregation in a software defined DSA prototype.

Even though spectrum allocation is studied earlier, simultaneous duplex communication for mesh network has not been studied for flow-based models. Observing that splitting spectrum into fixed bandwidth channels or allocating spectrum in per packet basis is not an optimal solution, a per session spectrum allocation was developed where a radio access spectrum depends upon its data rate requirements [10]. In this approach, each flow of data is considered as a session and each session

The authors acknowledge support from NSF award CNS #1346600.

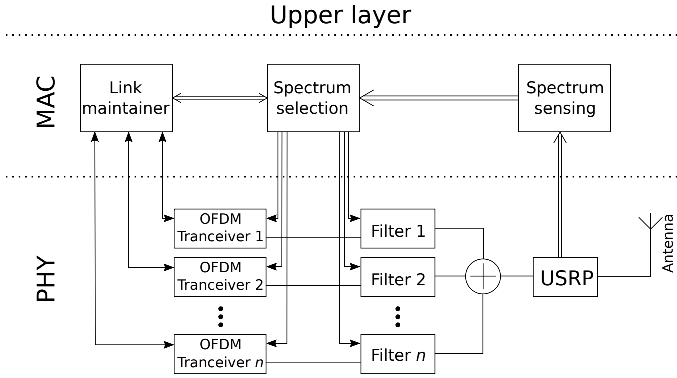


Fig. 2: System Architecture

can independently access spectrum. The technique first senses the available or free spectrum by sensing the power over a wide spectrum range and then detects spectrum opportunities using edge detection. Then several spectrum acquirement policies are analyzed. However, a node is limited to communicate only to a single neighbor at a time. In this paper, we present a framework where nodes are allowed to communicate with multiple nodes simultaneously. We first propose an algorithm which allows a group of nodes in a mesh network to negotiate for spectrum usage. Then we design a state-of-the-art prototype with off-the-shelf hardware which is capable of:

i) creating multiple heterogeneous virtual interfaces with a single radio interface;
ii) maintaining multiple independent simultaneous communications;
iii) transmission and reception simultaneously; and
iv) dynamically adapting the number of virtual interfaces and parameters of the virtual interfaces when the data rate requirement in a flow changes or PU arrives.

The experiment results confirm that the throughput of a mesh network can be enhanced up to 48% by using the dynamic adaptation of the spectrum for each flow.

The rest of the paper is organized as follow. In Section II we present the system architecture. Section III provides the details of the prototype developed with off the shelf SDRs. Section IV evaluates the performance of the proposed mechanisms and finally, Section V concludes the paper.

II. Methodology

The proposed design is shown in Figure 2 which exploits the power of SDR. We choose USRP as the hardware which is controlled with GNURadio. This architecture uses an array of OFDM transceiver blocks in present in GNURadio. Each of the OFDM transceiver blocks is connected to link maintenance blocks. A node maintains connections with all of its neighbors through different frequencies. Appropriate bandpass filters are applied to get rid of the spectrum leakage. The output of these filters are then added and the signal is transmitted to USRP. USRP then modulates this baseband signal to the chosen carrier frequency.

A. System assumptions

i) **Spectrum Fragmentation** The considered spectrum band is divided in N subcarriers. Here N can be 64, 128, 256, 512 or 1024. When a node wants to communicate with its neighbor, in our design we allocate a set of subcarriers to that link. The spectrum is divided into small fragments, and each fragment is associated with a link. The width of the fragment or the number of the subcarriers in a fragment is determined by the data-rate requirement of that particular link. The granularity of the fragment size depends on the number of subcarriers used for OFDM transmission. An increase in granularity enhances the robustness of the system but needs powerful digital signal processing capability. So, each hardware is limited with the total number of subcarriers and hence the granularity is limited.
ii) **Per session spectrum allocation** The design considers the subcarrier assignment to be dynamically changing with the throughput requirement of the links and the availability of spectrum. The current paper focuses on gossip based negotiation model where two hop neighbors collect the spectrum sensing map and decide on spectrum allocation to each link. The framework is presented in Section II-C.
iii) **OFDM and guard bands** In an ideal case, OFDM transmission should not emit energy on the subcarriers which are not allocated with data or pilot symbols. However due to improper design of filtering mechanism and transition bandwidth of filters, in practice, we need to provide guard bands on both sides of the spectrum fragments. After some trial and error, we have seen that leaving one subcarrier as guard band at both sides of a fragment is suitable for our testbed.
iv) **Heterogeneous channels** The channels are considered to be the spectrum allocated for each link. Since the fragments are assigned dynamically based on throughput requirements, they are of different size and inherently heterogeneous in nature.
v) **Full Duplex** Full duplex means a node can transmit and receive simultaneously. In our architecture, we allocate different subcarriers to each link. Since the fragments are not overlapping, a node can transmit and receive and receive data on nonoverlapping fragments. We have used two different antennas for transmission and reception in USRP B200 board.

B. Design challenges

During our system design we came across multiple challenges mainly related to hardware impairments. We list below the main challenges that we face during the system design:

i) **Interference isolation** We have encountered few kind of interferences. The first one is interference on signal reception by the transmitted signal of the same node. Although the transmitted signal and received signal are orthogonal in the frequency domain, there is still energy leakage. We eliminated this problem by separating the transmitter and receiver antenna and spacing them far apart. The second interference is co-channel interference. In this problem, if two links space very far apart also get the same subcarrier, cause interference on each other. This issue is solved by assigning the subcarriers that can not interfere. The third interference is cross-channel interference. Sometimes due to nonlinearity in the hardware, we experienced more cross-channel than expected. In that hardware, we widened the guard bandwidth.
ii) **Carrier frequency noise** Another interesting observation that we made is carrier frequency noise. In USRP radio the base band signal is modulated to the actual transmitting frequency. The central carrier frequency poses tremendous

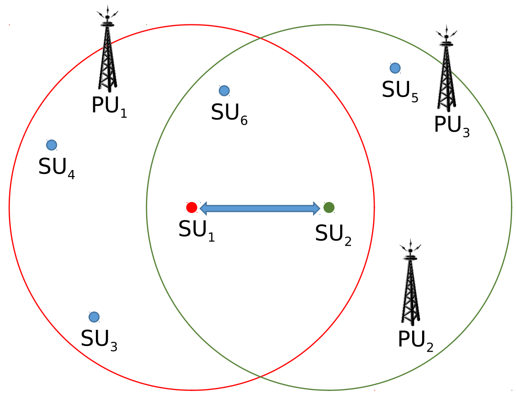


Fig. 3: An example scenario for node placement

noise on the particular subcarrier. While allocating data carriers for a baseband signal, we need to omit the subcarrier that overlaps with central frequency.

iii) **Preamble detection** Preambles are used for detection of transmission and synchronizing the transmitter and receiver. For very narrowband communication the preambles become an overhead. For three-way handshake, sometimes nodes have to send multiple preambles before the receiver can detect it.

iv) **Frequency offset in SDR** Due to hardware mismatch, the carrier frequency may be slightly mismatched. This may not seem any problem in most of the cases, but if the SDRs from a different manufacturer or they are of a different model then this problem can be hard to tackle when the transmissions are taking for a long time. We are yet to apply any solution for this issue. Periodic carrier synchronization seems a probable solution.

C. *Framework for dynamic spectrum acquirement*

In this section, we present the proposed method for spectrum allocation for a link. Figure 3 provides an example scenario of node placement. Here we are concentrating on the link between SU_1 and SU_2. As we can see, transmission of SU_1 will create interference to PU_1.

Each SU in our proposed framework follows the state diagram depicted in Figure 4. The receiver node will listen on the spectrum holes within its range. It will create a separate virtual interface for each of the holes and listen on the spectrum. When the transmitter is ready to transmit, it will start sending a beacon on one of the spectrum holes and try to complete three-way handshaking. If the three-way handshaking is not complete within an initialization cutoff, the transmitter will use a gossip-based protocol, where it asks other nodes to relay the handshaking request to the receiver, where both the nodes agree to a mutual spectrum for initial handshaking.

After the initial connection setup, the transmitter goes to stable state and keep transmitting on the approved spectrum. At a discrete time interval, both the transmitter and receiver sense the spectrum for PU arrival. If a portion of the used spectrum is blocked by a PU arrival, the transceiver pair communicates over the remaining agreed spectrum with lower data rate. They both scan for free spectrum and acquire the spectrum that is mutually open for both of them. However, if no such mutual spectrum is available, they request their neighbor to borrow spectrum. After that negotiation is over, they come to stable state again. Similar to PU arrival, if the data-rate demand of

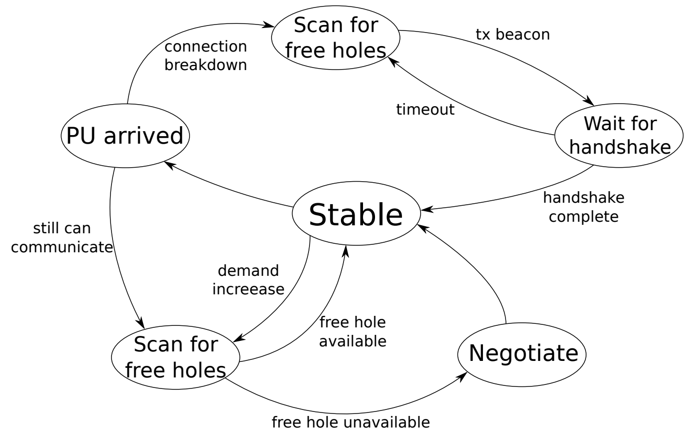


Fig. 4: State diagram of an SU

the transmitting node increases, the nodes follow the same procedure.

The negotiation for spectrum borrowing works on the principle that in multihop communication, not every node transmission will cause interference to all other nodes. For an example, in Figure 3, SU_4 is not causing any interference on SU_5. The node which wants to borrow spectrum from neighboring, request interference information from all neighbors and neighbors' neighbors. The neighbor nodes also send the usable frequencies. Now, a graph is formed where the links are converted to vertices and if two links are interfering, then an edge is drawn. A graph coloring algorithm is used to find out the optimal interfering spectrum block requirements. Then a reassignment of the spectrum is requested to the neighbors according to the new spectrum assignment. In the negotiation procedure, if the spectrum demand can not be fulfilled, each node will be assigned with spectrum proportional to their demand.

III. System Development

In our implementation and testing, we focus on two key points: 1) the prototype demonstrates desired performance; and 2) data loss of transmission and reception. We set up a small scale testbed experiment as shown in Figure 5. Each node consists of an USRP 200/210 board, fully integrated single board for signal processing, connected by USB 2.0 cable to a desktop computer running Ubuntu 14.04 operating system. The nodes are placed at a distance of 0.3 m from each other. All nodes function on a 2.442 GHz frequency with a 32 Khz sample rate. Figure 6 shows the overall physical setup.

Using GNURadio software development environment, we implement three different flow graphs corresponding to the transmitter node, the transceiver node, and the receiver node. In these flowgraphs, the prebuilt GNURadio Pad Sink and Pad Source blocks are just placeholders, giving us the flexibility to specify I/O data dynamically.

A. *Transceiver design*

Figure 7 and Figure 8 shows the GNURadio flow graph for the the transceiver node C (the relay node). Here, the figures represent the receiver and transmitter part respectively. The transmitting scheme consists of the following prebuilt GNURadio blocks: Stream to Tagged Stream, OFDM Transmitter, Multiply Const, Band Pass Filter, and UHD USRP Sink; the blocks' function is as follow.

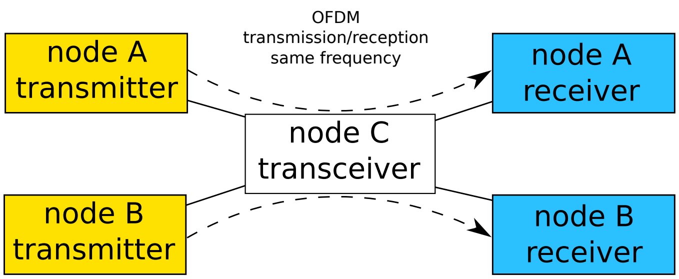


Fig. 5: Test Network Topology

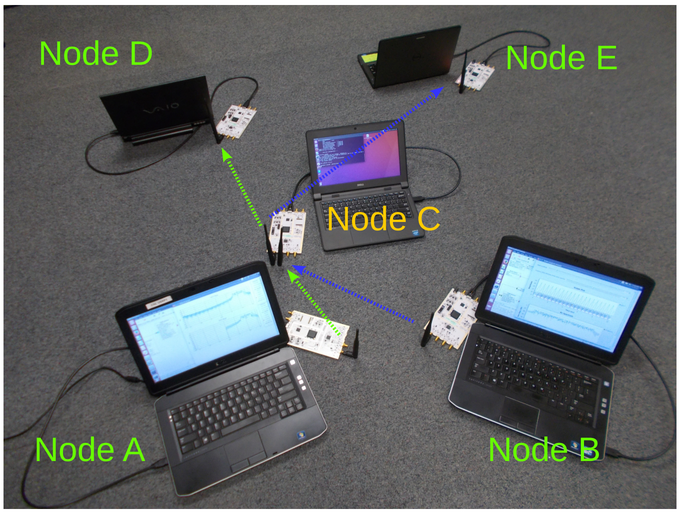


Fig. 6: Realtime transmission-reception setup with USRP boards

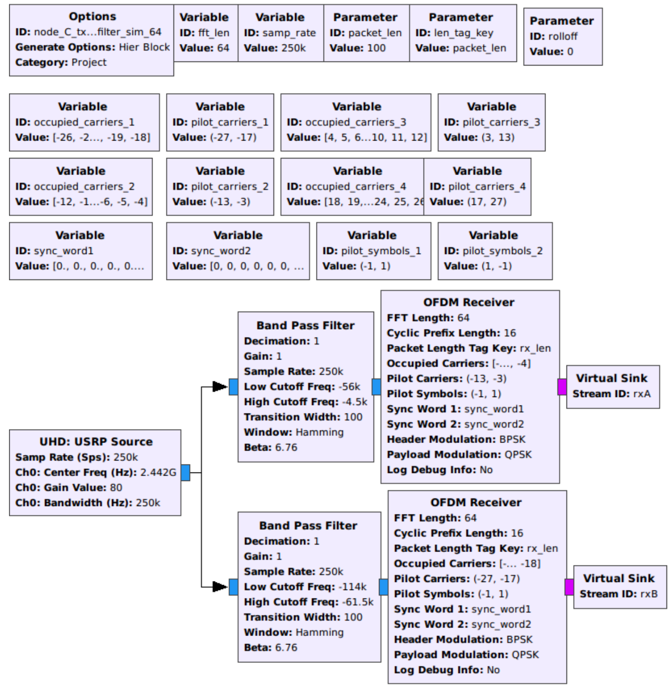


Fig. 7: The receive part of the relay node

1) **UHD: USRP source:** The USRP source block is required for receive signal using the USRP B200/210 board. The block captures the signal and output data stream based on the specified frequency, data output type, sample rate, and the antenna port.
2) **Band Pass Filter:** The block is imperative for parallel transmission on same band. Based on the OFDM transmission design, each carrier occupies 500 Hz (32KHz divided by 64). For each frame, the frequency cutoff will be 500 Hz multiplied by the value of smallest occupied carrier for low cutoff, and the largest value for high cutoff; an additional 500 Hz will be added on both side.
3) **OFDM receiver:** The OFDM Receiver is also a hierarchical block that handles several tasks. First, a Fast Fourier Transform (FFT) shifts the OFDM symbols into the frequency domain, where the signal processing is performed (the OFDM frame is in matrix form). Then, it is passed to a block that uses the preambles to perform channel estimation and coarse frequency offset, based on sync words. Both of these values are added to the output stream as tags; the preambles are then removed from the stream and not propagated. After that, both the coarse frequency offset correction and the equalizing are done in the Frame Equalizer block. The last block in the frequency domain is the Serializer block, which is the inverse block to the carrier allocator. It plucks the data symbols from the occupied carriers and outputs them as a stream of complex scalars, which will be converted to a byte stream.
4) **Stream to tagged stream:** The block will packetize the input byte stream from data input; it performs the necessary data preprocessing for the OFDM transmitter block.
5) **OFDM transmitter:** The OFDM Transmitter block is a hierarchical block, which handles both preprocessing and the OFDM transmission. In the preprocessing step, packetized byte stream produced by the Tagged Stream block will be trailed with CRC32, which reset to the new packet's length. CRC32 tagged stream in a byte will be repacked with a packet header and converted to the complex data stream, which concluded the preprocessing OFDM step. In the transmitting step, the complex data stream will have an OFDM frame allocated to it. The OFDM transmission requires the user to specify the packet length, the FFT length, the cyclic prefix length, the occupied carriers (the size of the OFDM frame), the pilot carrier, and the pilot symbol. We use the default value for sync words, header/payload modulation, roll-off-length, and scramble bits. We designed the OFDM transmission FFT length, the cyclic prefix length, the packet length, and the OFDM subcarriers.
6) **Multiply Const:** The USRP B200/210 board's processor will drop the packetized complex data stream if it detects an amplitude is increasing beyond a certain limit; we observed that the limit is 1. Therefore, the Multiply Const block is used to scale down the amplitude of the stream; trial, error, and correction show that the constant 0.045 give the best result.
7) **UHD: USRP Sink:** The USRP Sink block is required for transmitting signal using the USRP B200/210 board. The block takes in the generated signal and broadcasts it, based on the specified frequency, sample rate, and antenna port.

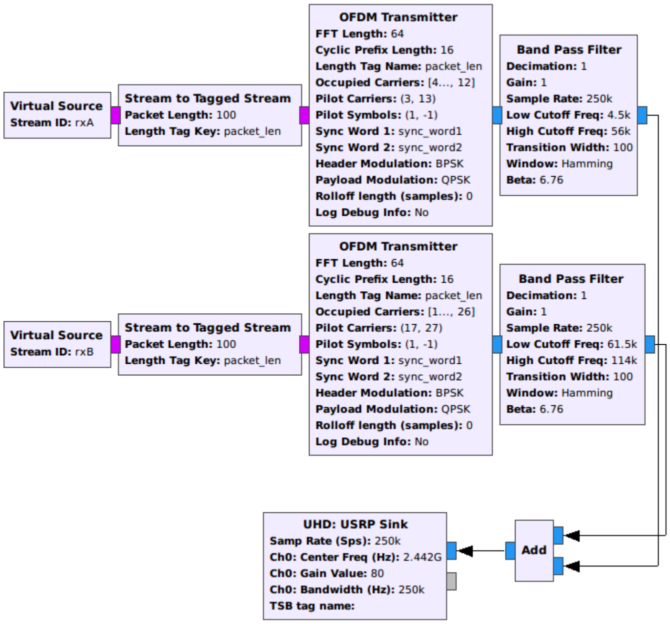


Fig. 8: The transmit part of the relay node

B. End node design

The end nodes, i.e. the traffic generating nodes and receiving nodes pose similar flowgraph as discussed earlier. However, the source node will have a **File Source** block which will read in the data file to be transmitted and output the data based on the user's choice of data type. We choose the output as byte and no repeat; the transmission will stop once the whole data file is processed and transmitted. Similarly, the destination node has a **File Sink** block instead of the virtual sink block. The block will write the received data to the output file.

IV. Results and Discussion

To test our prototype functionality, we set up two test scenarios: 1) Simulation with videos streaming; and 2) Real time transmission/reception with USRP boards.

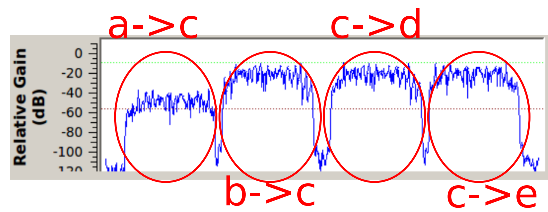
A. Experiment with Videos Streaming

In this scenario, we use VLC media player for streaming video and the GNURadio UDP Source and UDP Sink blocks. The same configuration is used as in Figure 5. In Node A, we use VLC to stream video captured by a webcam device. In Node B, we use VLC to stream video from an MP4 file. We replace the File Source and File Sink with UDP Source and UDP Sink blocks; they serve as a communication bridge between GNURadio and VLC. There is slight delay streaming between source and destination, but it is expected. The streaming quality is high; data loss is minimal. However, it is important to note that CPU usage is extremely high to perform the video streaming transmission/reception simultaneously with OFDM Transmitter and Receiver blocks.

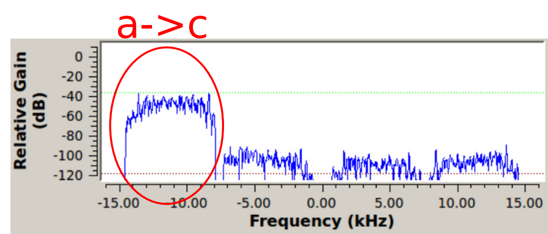
B. Real time Transmission/Reception with USRP boards

In the real-time transmission/reception scenario, we transmit an image file from Node A and B to Node C and have C relay the data reception to Node D and E. We observe that in real time transmission with USRP board, the designed bandpass filters frequency cutoff and gain value will interfere with the data transmission. Specifically, the gain value can cause signal distortion during transmission and reception. The bandpass filters frequency cutoff will cut off distorted signal with valid data packets. Furthermore, distorted signal can corrupt the data packets. The GNURadio OFDM Receiver block will drop any package that does not have a valid sync header value, which causes the data loss in reception.

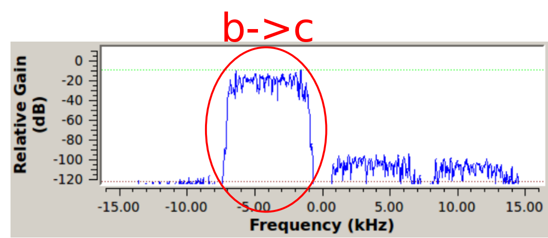
Figure 9 shows the FFT of received signal. Figure 9a represents the FFT of raw signal that is received at the USRP. This raw signal is passed through a band-pass filter to get the baseband signal coming from node A and B. The filtered signal coming from A and C can be seen in Figure 9b and Figure 9c respectively.



(a) FFT of received ray signal from USRP



(b) FFT of filtered signal from node A



(c) FFT of filtered signal from node B

Fig. 9: FFT of Real time signal transmission at the relay node

C. Results with varying simultaneous flows

The system is further evaluated for higher order OFDM subcarriers. The underlying schema is shown in Figure 10. Here a central node relays n data flows. Thus requiring $2n$ number of actual transmissions. Figure 11 and Figure 12 show the total throughput obtained by the system and the average throughput per flow respectively. Due to the guard carriers required at each side of the transmitter frequency blocks and needed number of subcarriers, the system with 64-bit subcarrier can not support more than two simultaneous flows (4 parallel transmissions). Similarly, 128 subcarriers can not support more than 4 streams. It is interesting to see that with an increase in subcarrier, the system throughput increases slightly. With the increase in a number of flows, the total throughput decreases linearly while the average throughput per flow decreases exponentially.

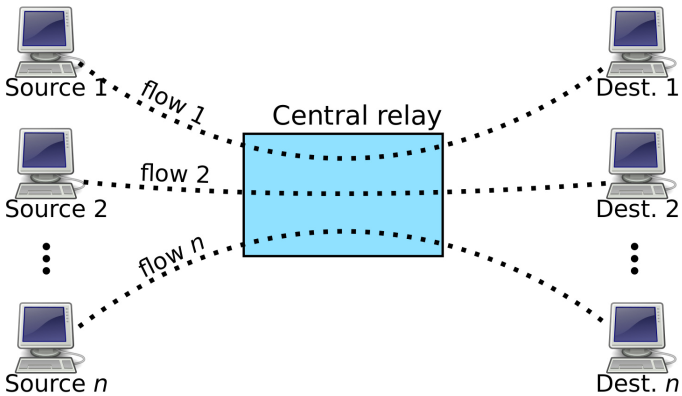


Fig. 10: Simulaton scenario

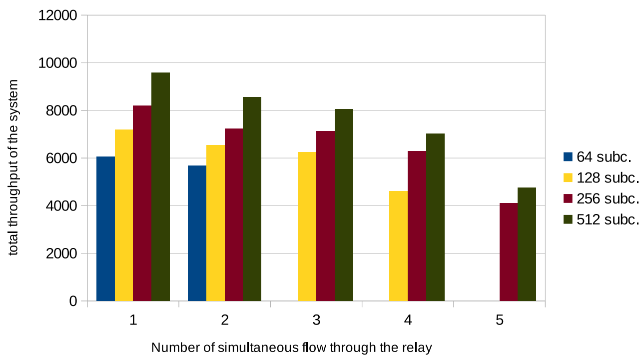


Fig. 11: Testbed Results for total throughput

D. Experiment with mesh network

In this section, we compare the proposed method presented in Section II-C with the conventional contention-based protocol. We have used the same network as depicted in Figure 3. PU arrival and departure process follow a normal distribution. The result is plotted in Figure 13. In our proposed mechanism, the nodes use proportional fairness for link width. Since the gossip-based protocol is used to borrow spectrum from exposed nodes, the total throughput decreases as the number of contesting nodes increase. However, the contention based protocol observe much steeper decrease due to the higher collisions. We observed up to 48% performance enhancement with our proposed framework.

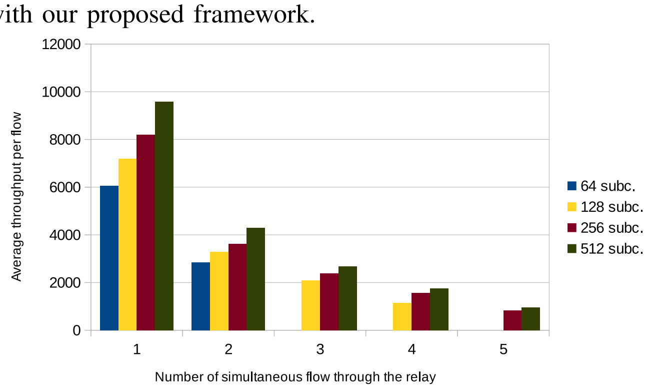


Fig. 12: Testbed Results for average throughput per flow

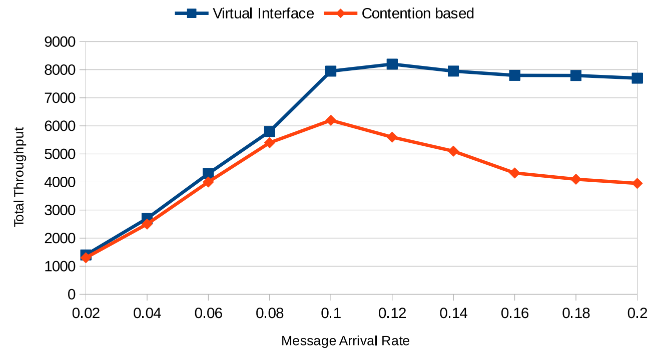


Fig. 13: Result for multiple transmission

V. Conclusion

In this paper, we presented a model for dynamic spectrum access in multihop networks. With a state-of-the-art testbed, we show that with a single radio interface a node can create multiple virtual interfaces to communicate with many neighbors simultaneously. A gossip-based framework is proposed to negotiate spectrum with neighbors and optimize the system performance. The system can achieve up to 48% enhancement for throughput over conventional contention based packet forwarding scheme.

References

[1] K. Chintalapudi, B. Radunovic, V. Balan, M. Buettener, S. Yerramalli, V. Navda, and R. Ramjee, “Wifi-nc: Wifi over narrow channels,” in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 4–4, USENIX Association, 2012.

[2] E. Magistretti, K. K. Chintalapudi, B. Radunovic, and R. Ramjee, “Wifi-nano: reclaiming wifi efficiency through 800 ns slots,” in *Proceedings of the 17th annual international conference on Mobile computing and networking*, pp. 37–48, ACM, 2011.

[3] Q. Zhao and B. M. Sadler, “A survey of dynamic spectrum access,” *Signal Processing Magazine, IEEE*, vol. 24, no. 3, pp. 79–89, 2007.

[4] E. Chen and C. Chu, “Channel estimation for nc-ofdm systems based on subspace pursuit algorithm,” in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, vol. 1, pp. 88–91, IEEE, 2012.

[5] L. Jiao, V. Pla, and F. Y. Li, “Analysis on channel bonding/aggregation for multi-channel cognitive radio networks,” in *Wireless Conference (EW), 2010 European*, pp. 468–474, IEEE, 2010.

[6] D. Chen, Q. Zhang, and W. Jia, “Aggregation aware spectrum assignment in cognitive ad-hoc networks,” in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, pp. 1–6, IEEE, 2008.

[7] F. Huang, W. Wang, H. Luo, G. Yu, and Z. Zhang, “Prediction-based spectrum aggregation with hardware limitation in cognitive radio networks,” in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pp. 1–5, IEEE, 2010.

[8] J. Lin, L. Shen, N. Bao, B. Su, Z. Deng, and D. Wang, “Channel characteristic aware spectrum aggregation algorithm in cognitive radio networks,” in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pp. 634–639, IEEE, 2011.

[9] S. Bhunia, V. Behzadan, and S. Sengupta, “Enhancement of spectrum utilization in non-contiguous dsa with online defragmentation,” in *Military Communications Conference, MILCOM 2015-2015 IEEE*, pp. 432–437, IEEE, 2015.

[10] L. Yang, Z. Zhang, W. Hou, B. Y. Zhao, and H. Zheng, “Papyrus: a software platform for distributed dynamic spectrum sharing using sdrs,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 31–37, 2011.