Routing and Dynamic Spectrum Allocation for Cognitive Radio Networks

Pradeep
MTech in Software Engineering, Department of ISE, PESIT, Bangalore

ABSTRACT

Cognitive Radio Networks (CRNs) are mainly used for path selection due to the added advantage of dynamic spectrum utilization when the frequency band is not under utilization. Path selection in multi-hop cognitive radio networks is done using dynamic spectrum allocation. We show that the overall throughput and the packet drop ratio determined by the proposed route selection strategy are better than the existing methods or strategies. The main aim behind our approach is to guarantee that the link failure is minimized and throughput is maximized. A polynomial time algorithm is designed for this problem and evaluated using Network Simulator for computation of simulation results. The results show that our proposed algorithm achieves a near optimal solution of this problem for multi-hop CRNs.

General Terms

Ad-hoc On-demand Distance Vector(AODV) protocol,

Polynomial time algorithm, Route robustness, Routing table, Spectrum utilization

Keywords

Cognitive radio, Dynamic spectrumallocation, Primary users, Secondary users, Spectrum hole, Spectrum pooling, Weighted Cumulative Expected Transmission Time(WCETT).

1. Introduction and related work

Cognitive Radio (CR)[1] is a technique which allows secondary users to access the licensed spectrum when no primary users are using the spectrum only under the condition that the normal operation of primary users will not be interrupted. This technique allows for the dynamic access of spectrum unlike other ad-hoc networks and hence provides better performance.

Spectrum holes[2],[3] refer to those range of spectrum bands which are not utilized by any of the licensed users. These spectrum holes are the range of frequency bands which are utilized by the secondary users. Since this process is done based on the availability of spectrum, it is dynamic in nature and hence referred to as dynamic spectrum utilization.

Spectrum pooling[4] is a concept used in CRNs. Suppose a secondary user needs 10 MHz of bandwidth but the contiguous available bandwidth is just 3 MHz. In such cases, if 10 MHz is available but not contiguously then it is pooled or combined into a single block and given to the secondary user to satisfy the particular requirement. This process is referred to as spectrum pooling.

The basic cognitive cycle[5],[6] consists of the following steps:

Step 1: Detection of Radio Frequency stimuli which consists of detecting the spectrum holes available, estimating the channel state information and predicting the channel capacity usage.

Step 2: Spectrum management[7] which includes traffic shaping, routing and providing Quality of Service to the users.

Step 3: The final step is to put these into action by means of transmission with minimum power capabilities and minimum packet loss due to disruptions such as node or link failures in the network.

2. Methodology

In this paper, to enable CR features Weighted Cumulative Expected Transmission Time (WCETT)protocol[8] is used apart from Ad-hoc On-demand distance Vector (AODV) protocol[9]. In AODV from the source node packets are transferred to all the neighboring nodes. Each of these nodes then transfer the packets to their neighboring nodes until the packet reaches the destination.

The routing table[10] stores the information of the neighboring nodes only. As soon as a node receives a packet it forwards the packet to all it's neighbors. The neighbors' information is stored in the routing table. The routing table keeps changing dynamically.

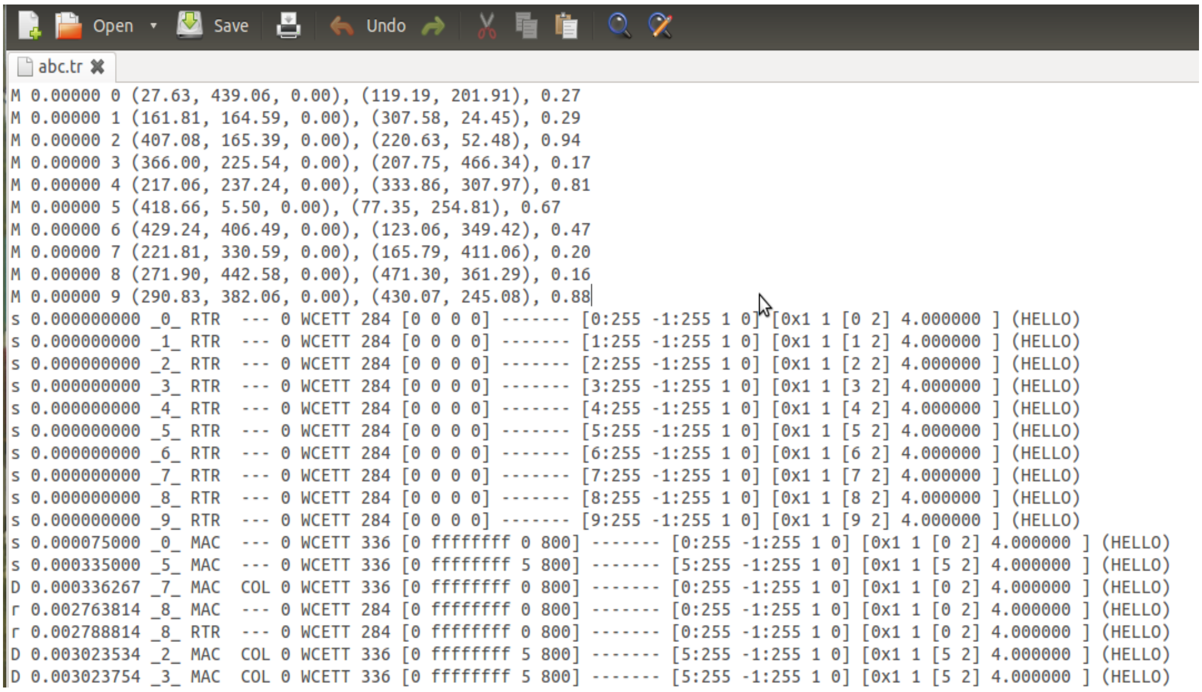
We use a variation of AODV protocol i.e the reply packets need not follow the same path as request packets. The existing AODV protocol allows the nodes to choose a different route if at all problems of congestion occur but they do not allow to choose a different path for acknowledgement. They have to follow the request packet path only. Hence if congestion problems occur during reply packet transfer, the nodes have to wait leading to latency and delay problems. Hence we use the variation of AODV where reply packets can choose a different path to acknowledge the source node. This increases the throughput because while transferring the request packets the path maybe free but during transfer of reply packets the path maybe busy. By choosing a different path for transfer of reply packets we overcome the waiting time and hence better performance.

3. Packet format and Routing table

The packet format for all kind of packets of WCETT protocol must be defined in the header file. The main kinds of packets are HELLO[11] for route discovery, RREQ for route request, RREP for successful route reply and RERR for unsuccessful route reply. The formats for each of these are defined in a separate structure.

The output of simulation is stored in a trace file. The trace file gives various information such as positions of the nodes, packets sent or received or dropped, sequence number of the packets, destination port number, header format, type of the packets sent. By analyzing the trace file we can get various details regarding the packet transfer. These values are used to plot the final graph of the simulation results.

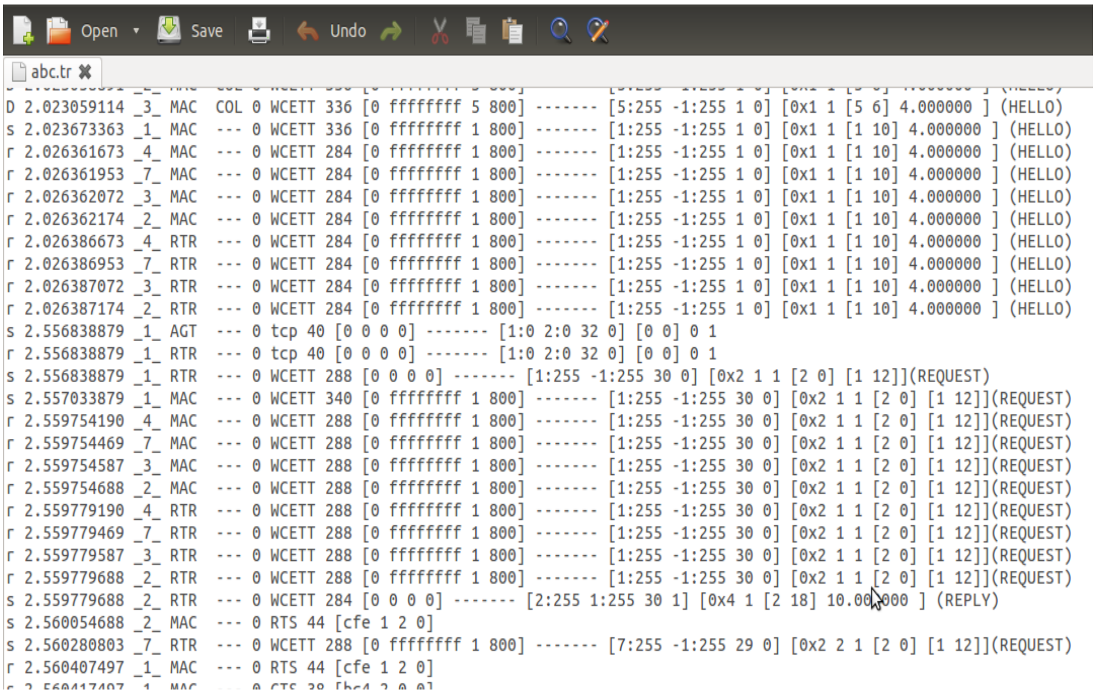
Figure 3.1 shows a part of the trace file. Here in the first column M represents a mobile node. Numbers 0 to 9 in the third column represents the node number. The positions of each of these nodes is specified as (x,y) in the co-ordinate system. In the first column, 's' represents packets sent, 'D' represents packets dropped, 'r' represents packets received. The third column (_x_) represents the sequence number of packets. The sequence number must be unique so that packets can be identified at the destination and duplicates can be dropped. It is also useful for reassembling the fragmented packets at the receiver. In the fourth column, RTR represents routing packet, MAC represents link layer packet, PHY represents physical layer packet, AGT represents agent packet. The seventh column represents the protocol used and hence is seen as WCETT in Figure 3.1. The eighth column represents the destination port address of the node. Next column represents the flag values. The last column represents the type of packet and here in Figure 2.1 it is HELLO packet. HELLO packet is sent for route discovery, if successfully transmitted and acknowledgement received then we decide that a route exists and can send the REQUEST[12] packet subsequently.




```
M 0.00000 0 (27.63, 439.06, 0.00), (119.19, 201.91), 0.27
M 0.00000 1 (161.81, 164.59, 0.00), (307.58, 24.45), 0.29
M 0.00000 2 (407.08, 165.39, 0.00), (220.63, 52.48), 0.94
M 0.00000 3 (366.00, 225.54, 0.00), (207.75, 466.34), 0.17
M 0.00000 4 (217.06, 237.24, 0.00), (333.86, 307.97), 0.81
M 0.00000 5 (418.66, 5.50, 0.00), (77.35, 254.81), 0.67
M 0.00000 6 (429.24, 406.49, 0.00), (123.06, 349.42), 0.47
M 0.00000 7 (221.81, 330.59, 0.00), (165.79, 411.06), 0.20
M 0.00000 8 (271.90, 442.58, 0.00), (471.30, 361.29), 0.16
M 0.00000 9 (290.83, 382.06, 0.00), (430.07, 245.08), 0.88
s 0.000000000 _0_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [0:255 -1:255 1 0] [0x1 1 [0 2] 4.000000 ] (HELLO)
s 0.000000000 _1_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [1:255 -1:255 1 0] [0x1 1 [1 2] 4.000000 ] (HELLO)
s 0.000000000 _2_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [2:255 -1:255 1 0] [0x1 1 [2 2] 4.000000 ] (HELLO)
s 0.000000000 _3_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [3:255 -1:255 1 0] [0x1 1 [3 2] 4.000000 ] (HELLO)
s 0.000000000 _4_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [4:255 -1:255 1 0] [0x1 1 [4 2] 4.000000 ] (HELLO)
s 0.000000000 _5_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [5:255 -1:255 1 0] [0x1 1 [5 2] 4.000000 ] (HELLO)
s 0.000000000 _6_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [6:255 -1:255 1 0] [0x1 1 [6 2] 4.000000 ] (HELLO)
s 0.000000000 _7_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [7:255 -1:255 1 0] [0x1 1 [7 2] 4.000000 ] (HELLO)
s 0.000000000 _8_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [8:255 -1:255 1 0] [0x1 1 [8 2] 4.000000 ] (HELLO)
s 0.000000000 _9_ RTR  --- 0 WCETT 284 [0 0 0 0] ------- [9:255 -1:255 1 0] [0x1 1 [9 2] 4.000000 ] (HELLO)
s 0.000075000 _0_ MAC  --- 0 WCETT 336 [0 ffffffff 0 800] ------- [0:255 -1:255 1 0] [0x1 1 [0 2] 4.000000 ] (HELLO)
s 0.000335000 _5_ MAC  --- 0 WCETT 336 [0 ffffffff 5 800] ------- [5:255 -1:255 1 0] [0x1 1 [5 2] 4.000000 ] (HELLO)
D 0.000336267 _7_ MAC  COL 0 WCETT 336 [0 ffffffff 0 800] ------- [0:255 -1:255 1 0] [0x1 1 [0 2] 4.000000 ] (HELLO)
r 0.002763814 _8_ MAC  --- 0 WCETT 284 [0 ffffffff 0 800] ------- [0:255 -1:255 1 0] [0x1 1 [0 2] 4.000000 ] (HELLO)
r 0.002788814 _8_ RTR  --- 0 WCETT 284 [0 ffffffff 0 800] ------- [0:255 -1:255 1 0] [0x1 1 [0 2] 4.000000 ] (HELLO)
D 0.003023534 _2_ MAC  COL 0 WCETT 336 [0 ffffffff 5 800] ------- [5:255 -1:255 1 0] [0x1 1 [5 2] 4.000000 ] (HELLO)
D 0.003023754 _3_ MAC  COL 0 WCETT 336 [0 ffffffff 5 800] ------- [5:255 -1:255 1 0] [0x1 1 [5 2] 4.000000 ] (HELLO)
```

Figure 3.1: HELLO packets in the trace file

Figure 3.2 shows a part of the trace file[13] with REQUEST packets. From the figure we can observe the packet type is REQUEST in the last column. Once HELLO packets are sent, REQUEST packets are sent to those nodes for which a link exists. The REQUEST packets are sent to all the neighboring nodes which in turn are transferred to their corresponding neighbors. The REQUEST packet can be successfully transmitted or else dropped. Dropped packets maybe retransmitted via a different path if exists. On receiving the REQUEST packet, the nodes send an acknowledgement to the sender or source.



```
D 2.023059114 _3_ MAC COL 0 WCETT 336 [0 ffffffff 5 800] ------- [5:255 -1:255 1 0] [0x1 1 [5 6] 4.000000 ] (HELLO)
s 2.023673363 _1_ MAC --- 0 WCETT 336 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026361673 _4_ MAC --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026361953 _7_ MAC --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026362072 _3_ MAC --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026362174 _2_ MAC --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026386673 _4_ RTR --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026386953 _7_ RTR --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026387072 _3_ RTR --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
r 2.026387174 _2_ RTR --- 0 WCETT 284 [0 ffffffff 1 800] ------- [1:255 -1:255 1 0] [0x1 1 [1 10] 4.000000 ] (HELLO)
s 2.556838879 _1_ AGT --- 0 tcp 40 [0 0 0 0] ------- [1:0 2:0 32 0] [0 0] 0 1
r 2.556838879 _1_ RTR --- 0 tcp 40 [0 0 0 0] ------- [1:0 2:0 32 0] [0 0] 0 1
s 2.556838879 _1_ RTR --- 0 WCETT 288 [0 0 0 0] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
s 2.557033879 _1_ MAC --- 0 WCETT 340 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559754190 _4_ MAC --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559754469 _7_ MAC --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559754587 _3_ MAC --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559754688 _2_ MAC --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559779190 _4_ RTR --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559779469 _7_ RTR --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559779587 _3_ RTR --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
r 2.559779688 _2_ RTR --- 0 WCETT 288 [0 ffffffff 1 800] ------- [1:255 -1:255 30 0] [0x2 1 1 [2 0] [1 12]](REQUEST)
s 2.559779688 _2_ RTR --- 0 WCETT 284 [0 0 0 0] ------- [2:255 1:255 30 1] [0x4 1 [2 18] 10.000000 ] (REPLY)
s 2.560054688 _2_ MAC --- 0 RTS 44 [cfe 1 2 0]
s 2.560280803 _7_ RTR --- 0 WCETT 288 [0 ffffffff 1 800] ------- [7:255 -1:255 29 0] [0x2 2 1 [2 0] [1 12]](REQUEST)
r 2.560407497 _1_ MAC --- 0 RTS 44 [cfe 1 2 0]
```

Figure 3.2: REQUEST packets in the trace file

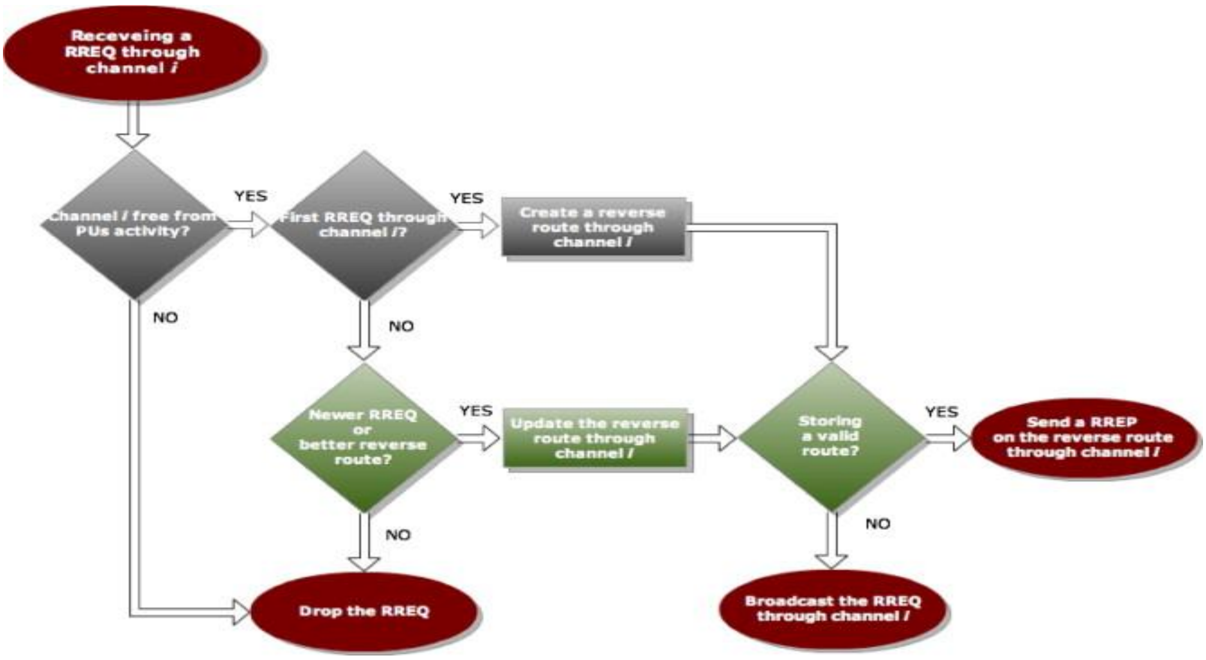


Figure 3.3: Flowchart

4. Simulation results and discussion

We create 10 mobile nodes. At time 0 s there is no packet transfer and hence only the nodes can be seen. Here nodes 1 and 2 are primary nodes. The rest of the 8 nodes are secondary.

The connection between the primary nodes 1 and 2 is initially established using Transmission control Protocol (TCP).The packet transfer happens as shown in Figure 4.1. Since TCP is used it is connection oriented. The spectrum is allocated to the primary users 1 and 2. When they are idle it can be utilized by the rest of the nodes dynamically.Packets are transferred between nodes 1 and 2 continuously. Primary users have greater priority than secondary users. Secondary users can utilize the spectrum only when primary users are idle. Else the spectrum is first allocated to primary nodes since they are licensed users.

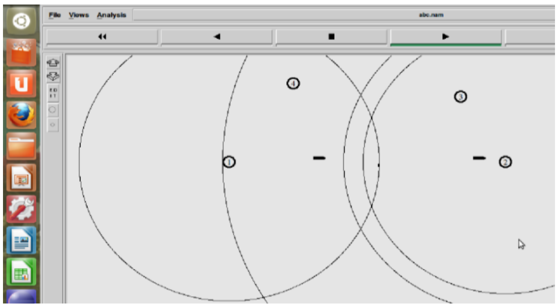


Figure 4.1: Communication between primary users

After utilizing the spectrum once nodes 1 and 2 become idle, the request of secondary user 7 is accomplished and packets are transferred between nodes 7 and 8 as shown in Figure 4.2. These nodes can utilize the spectrum until another primary user requests for the spectrum. Secondary users use User Datagram Protocol (UDP) for communication because secondary users access the spectrum dynamically for a short duration of time and hence it would be time consuming to establish connection using TCP by means of three-way handshake.

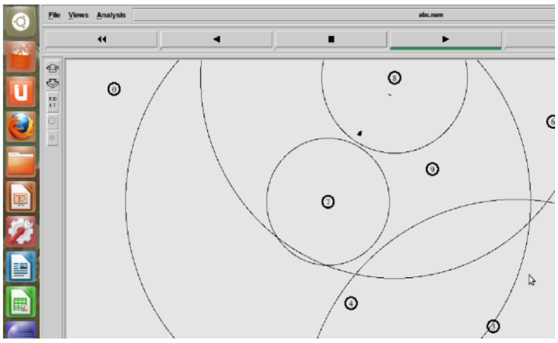


Figure 4.2: Communication between secondary users

After the entire transfer of packets for a given time duration we monitor the system and find the number of packets dropped during transmission. The packets dropped are stored in a separate queue. The packet drop ratio is as shown in Figure 4.3.

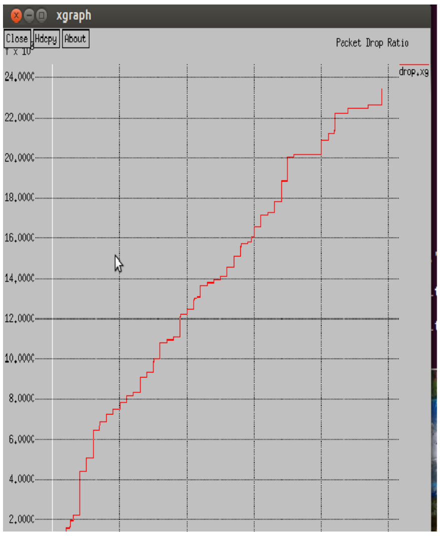


Figure 4.3: Graph of packet drop ratio

After the entire transfer of packets for a given time duration we monitor the system and find the number of packets successfully transferred during transmission. The throughput is as shown in Figure 4.4. The throughput reaches a peak value of around 15,000 and again drops down and remains constant for the entire time duration. But from the graph we can infer that the throughput remains high and almost the same throughout thus proving the efficiency of our algorithm.

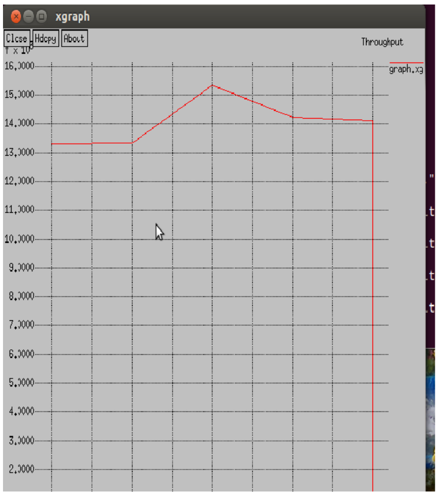


Figure 4.4: Graph of throughput

The simulation results are tabulated as shown in Table 4.1.

Time	Throughput	Packet drop ratio
0	0	0
10	4098	0.732
20	5624	0.754
30	6368	0.802
40	7098	0.881
50	8064	0.9012

Table 4.1: Simulation results

5. Conclusion

In this paper, we consider routing and spectrum allocation problem in multi-hop cognitive radio networks. We take into account the channel heterogeneity property and the channel dynamics of CRNs. We show that route robustness greatly impacts system performance.The routes chosen and spectrum allocated on each link along these routes is chosen such that the system throughput is maximized and number of packet drops is minimized. The aim behind our solution is to guarantee a basic level of robustness for a set of routes based on which routes are selected and the channel on each link along the routes isallocated. We also propose an algorithm with polynomial time complexity such that the execution time does not greatly increase even when the number of nodes are increased. The performance of the proposed mechanism is evaluated via simulations. The results show that the solution obtained by the polynomial time complexity algorithm is near optimal and can achieve a good balance between performance and time complexity. The results are tabulated by means of simulation using NS-2.3.1.

ACKNOWLEDGEMENTS

I would like to express my immense gratitude to Prof.Shylaja S S**,** Professor and Head of Department of ISE, PES Institute of Technology, Bengaluru, for her help and inspiration during the tenure of the course.

I would also express my thanks and gratitude to my Internal guide Ms.Sangeetha J, Associate Professor, Department of ISE, PES Institute of Technology, Bengaluru, for her help and inspiration during the entire duration of the project work.

REFERENCES

[1] Chao-Fang Shih, Wanjiun Liao, and Hsi-Lu Chao, “Joint Routing and Spectrum Allocation for Multi-Hop Cognitive Radio Networks with Route Robustness Consideration”, IEEE Transactions on wireless communications, vol. 10, no. 9, september 2011

[2] T. Fujii and Y. Yamao, “Multi-band routing for ad-hoc cognitive radio networks,” IEEE SDR, 2006.

[3] M. Sharma, A. Sahoo, and K. D. Nayak, “Channel selection under interference temperature model in multi-hop cognitive mesh networks,” IEEE DySPAN, 2007.

[4] H. Khalife, S. Ahuja, N. Malouch, and M. Krunz, “Probabilistic path selection in opportunistic cognitive radio networks,” IEEE Globecom, 2008.

[5] Y. R. Kondareddy, P. Agrawal, and K. Sivalingam, “Cognitive radio network setup without a common control channel,” IEEE MILCOM, 2008.

[6] C.-F. Shih and W. Liao, “Exploiting route robustness in joint routing and spectrum allocation in multi-hop cognitive radio networks,” IEEE WCNC, 2010.

[7] H. Kushwaha, Y. Xing, R. Chandramouli, and H. Heffes, “Reliable multimedia transmission over cognitive radio networks using fountain codes,” Proc. IEEE, 2008.

[8] C. Peng, H. Zheng, and B. Y. Zhao, “Utilization and fairness in spectrum assignment for opportunistic spectrum access,” ACM Mobile Networks and Applications, 2006.

[9] H. Zheng and C. Peng, “Collaboration and fairness in opportunistic spectrum access,” IEEE ICC, 2005.

[10] H. Zheng and L. Cao, “Device-centric spectrum management,” IEEE DySPAN, 2005.

[11] M. Thoppian, S. Venkatesan, R. Prakash, and R. Chandrasekaran, “Maclayer scheduling in cognitive radio based multi-hop wireless networks,” IEEE WoWMoM, 2006.

[12] L. Cao and H. Zheng, “Distributed spectrum allocation via local bargaining,” IEEE SECON, 2005.

[13]http://www.isi.edu/nsnam/ns/ns-documentation