**Spectrum Profitability based Channel Allocation for reducing system overhead in Cognitive Radio**

**Mrs. M. Vaidehi \*1, Mrs. D. Umamaheshwari 2, Mary Amala Jenni 3**

*1 Assistant Professor, Department of ECE, St.Anne’s CET, Anguchettypalayam,Panruti,India*

*2 Assistant Professor, Department of ECE, St.Anne’s CET, Anguchettypalayam,Panruti,India*

*3 Assistant Professor, Department of ECE, St.Anne’s CET, Anguchettypalayam,Panruti,India*

**Abstract**

***In older days, the spectrum is divided into portions and each portion is used by separate organizations. This is called static spectrum allocation. But, the organizations do not use the spectrum, fully both in case of time and in the case of bandwidth. The scarce resource spectrum is wasted. So, dynamic spectrum allocation is introduced. In this technique, the communication is done through the bandwidth which is free. So, wastage of bandwidth is reduced partially.***

1. **Introduction**

Cognitive (or smart) radio networks are an innovative approach to wireless engineering in which radios are designed with an unprecedented level of intelligence and agility. This advanced technology enables radio devices to use spectrum (i.e., radio frequencies) in entirely new and sophisticated ways. Cognitive radios have the ability to monitor sense and detect the conditions of their operating environment, and dynamically reconfigure their own characteristics to best match those conditions.

The main scope of the work is to maximize the profitability of network using the combined framework of routing and channel allocation. The profit of a network is determined by the difference between the cost of serving a request and the revenue obtained from serving that request. In order to increase the profitability, the physical cost of the system to compute the route and to make decisions about the spectrum allocation should be reduced. In this paper an economic framework for reducing the complexity of routing and channel allocation is proposed.

1. **Problems in Existing System**

In the existing system, the channel allocation and routing are two separate processes and both has high overhead. They require complex computations to be performed. So overhead of the system is increased. In the existing system, the connection of the secondary user is terminated at once the primary user is online. So, reward for serving the secondary user node will not be obtained. Also in systems like multi hop routing will not assure uninterrupted channel availability for primary users. The algorithm used in the existing system is based on the architecture of the network. The two types are centralized and decentralized. The algorithm used for centralized architecture does not suit decentralized network and vice versa. The shadow price of edges is taken into account in the existing algorithm for routing. But the edge price will fail in the arrival of the primary user. The channel availability for primary users is affected due to the wrong decision on spectrum allocation by the base station.

1. **Proposed System**

In the proposed system, the routing and channel allocation is done combined. The complexity of the proposed algorithm is low when compared to the existing algorithms. The algorithm is mainly based on the shadow price of nodes not the edges. So, channel allocation and routing are effective. It assures the uninterrupted channel availability for primary users. It checks the capacity of the channel before rerouting the secondary user connection. If the capacity is enough for the primary user and secondary user connection, the there is no rerouting. The usability of spectrum is reasonably improved when compared to the existing system. Also the status of primary user is accurately detected by simple automated entry and exit notification so that the channel availability for primary user is certain. The algorithm responds equally for networks of different size. It does not require any prior information other than the shadow price of nodes which is the primary input of the algorithm. The status of the network is tracked and this information will be useful for speeding up the algorithm.

1. **System Architecture**

A simple architecture with one primary user, one secondary user and one router is used in this work as shown in Fig.1.



**Figure 1: System Architecture**

The router allocate channel for the primary user without any interrupt but for the secondary user it will allocate based on availability of spectrum.

1. **Use case diagram**

Use case diagram is the primary system model that will show the user interaction of the user with the system. Figure 2 shows the use case diagram of the proposed work as shown in the figure router handle spectrum allocation for the primary and secondary user and also the primary user interrupt to the secondary user allocated spectrum apart from the routing task.

1. **Proposed Algorithm: Dijkstra Algorithm with Node Shadow Price**

Let the node at which we are starting to be called the initial node. Let the distance of node *Y* is the distance from the initial node to *Y*. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Mark all nodes unvisited. Set the initial node as current. Create a set of the unvisited nodes called the unvisited set consisting of all the nodes.
3. For the current node, consider all of its unvisited neighbours and calculate their tentative distances. Compare the newly calculated tentative distance to the current assigned value and assign the smaller one. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbour B has length 2, then the distance to B (through A) will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.



**Figure 2: Use Case Diagram Of Proposed System**

1. **Proposed Algorithm: Dijkstra Algorithm with Node Shadow Price**

Let the node at which we are starting to be called the initial node. Let the distance of node *Y* is the distance from the initial node to *Y*. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Mark all nodes unvisited. Set the initial node as current. Create a set of the unvisited nodes called the unvisited set consisting of all the nodes.
3. For the current node, consider all of its unvisited neighbours and calculate their tentative distances. Compare the newly calculated tentative distance to the current assigned value and assign the smaller one. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbour B has length 2, then the distance to B (through A) will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.
4. When we are considering all of the neighbours of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.
6. Select the unvisited node that is marked with the smallest tentative distance, and set it as the new "current node" then go back to step 3.

The performance of the algorithm is evaluated by increasing the no of nodes in the network. Figure 4 shows that the packet delivery of the proposed scheme at 2 node alone is around 99% if we increase the no of node the successful delivery ratio is decreases with increased propobablity of interrupts by the primary user.at worst case at 60 nodes the packet delivery radio is around 75%.

End to end delivery delay is another performance metric for the routing algorithm. figure 5 shows the secondary user packet delivery delay of the secondary user with the increased no of node in the network. From the graph we can observe that at worst case the proposed algorithm provide around 50ms delay when no of node is 60.

**Function** Djikstra (*Graph*,*source*):

dist. [*source*] := 0 *// Distance from source to source* **for each** vertex*v*in*Graph*:*// Initializations*

**if***v*≠*source*

dist[*v*] := infinity *// Unknown distance function from source to* *v*

previous[*v*] := undefined *// Previous node in optimal path* *from source*

**end if**

add *v* to *Q// All nodes initially in Q (unvisited nodes)* **end for**

**while***Q***is not** empty:*// The main loop*

*u* := vertex in *Q* with min dist[u] *// Source node in first case* remove *u* from *Q*

**for each** neighbor*v*of*u*:*// where v has not yet been removed**from Q.*

*alt* := dist[*u*] + length(*u*, *v*)

**if***alt*< dist[*v*]:*// A shorter path to v has been found*dist[*v*] := *alt*

previous[*v*] := *u* **end if**

**end for end while**

**return** dist[], previous[] **end function**

**Figure 3:** Pseudo Code of proposed algorithm



 **Figure 4:** Secondary user packet **Figure 5:** Secondary user

 packet delivery radio vs no of nodes delivery delay vs no of

 in the network. nodes in the network

1. **Conclusion and Future Work**

This work finds the most efficient path to travel the packets over a network. It allocates the bandwidth for both primary user and secondary user. The algorithm does not consider Byzantine attacks. But it can be controlled by integrating adaptive reputation based clustering in decision making. The usability of spectrum is not increased to hundred percent. Once the decision is made, other secondary users would not get channel though they have equal priority. To avoid this, scheduling can be done, but it will increase the overhead of the system. A simple scheduling algorithm that should not increase the overhead at the same time, it should not make the nodes to starve in the queue.

**Reference**

1. FCC Spectrum Policy Task Force. Report of the spectrum efficiency working group. Fed Commun Washington DC. Tech Rep ET Docket no. 02-135; November 2002.
2. Cabric D, Mishra S, Brodersen R. Implementation issues in spectrum sensing for cognitive radios. In: Proc Asilomar conf on signals, systems and computers; November 2004. p. 772-6.
3. A. Ghasemi, E.S. Sousa, Collaborative spectrum sensing for opportunistic access in fading environments, in: IEEE Symposium, New Frontiers in Dynamic Spectrum Access Networks (DySPAN05), Baltimore, USA, 2005, pp. 131-136
4. Akyildiz IF, Lee WY, Vuran MC, Mohanty S. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey. J Comput Netw (Elsevier) 2006;50:2127-59.
5. S.M. Mishra, A. Sahai, R. Brodersen, Cooperative sensing among cognitive radios, in: IEEE International Conference in Communications (ICC06), Turkey, vol. 4, 2006, pp. 1658-1663.
6. G. Ganesan, Y.G. Li, Cooperative spectrum sensing in cognitive radio part II: multiuser networks, IEEE Trans. Wireless Commun. 6 (6) (2007) 2214-2222.
7. Y. Chen, Optimum number of secondary users in collaborative spectrum sensing considering resources usage efficiency, IEEE Commun. Lett. 12 (12) (2012).
8. A. Mesodiakaki, F. Adelantado, L. Alonso, C. Verikoukis, Energy-efficient contention-aware channel selection in cognitive radio ad-hoc networks, in: Proceedings of the IEEE CAMAD, 2012.