Channel Assignment Algorithms: A Comparison of Graph Based Heuristics

Husnain Mansoor Ali
University Paris Sud 11
Centre Scientifique d'Orsay
91405 Orsay - France
husnain.ali@u-psud.fr

Anthony Busson
University Paris Sud 11
Centre Scientifique d'Orsay
91405 Orsay - France
anthony.busson@u-psud.fr

Véronique Vèque
University Paris Sud 11
Centre Scientifique d'Orsay
91405 Orsay - France
veronique.veque@u-psud.fr

ABSTRACT

This paper gives the comparison of different channel assignment heuristics proposed in the literature and introduces a new algorithm named MCAIR. Specifically, it compares, static, multi-radio multi-channel algorithms which are graph theory based and where a priori traffic characteristics are unknown. It also proposes new metrics other than overall interference which guide in better evaluating the assignment.

Keywords

Multi-Radio, Channel Assignment, Algorithms, Comparison, Ad hoc Networks

1. INTRODUCTION

Cost effective technologies like 802.11 [1] have changed the way we use communications and computing. Due to its success and wide-spread use, the spectrum resources allocated for it is becoming more and more crowded. The IEEE 802.11 a and b/g provides number of non overlapping/orthogonal channels (12 and 3 respectively) which can be used simultaneously. Number of studies have been done which propose the simultaneous utilization and automatic configuration of nodes to use more channels to provide minimum interference and thus maximize throughput([4], [2], [11], [16], [17], [12], [18]). The focus of most of these studies have been on configuration of access points or mesh nodes. Thus different Channel Assignment(CA) strategies have been proposed based on graph theoretic approaches, linear programming or simple common sense heuristics which try to achieve this purpose. [13] gives a taxonomy of different CA schemes.

Capacity increase is the main motivation for using multiple radios. P. Gupta and P. R. Kumar's [6] pioneering work provided bounds on maximum capacity that can be achieved by a single channel wireless networks. For multiple channels these bounds do not hold true as multiple channels provide additional capacity along with constraints. In [8], the authors calculate the capacity of multi-channel multi-radio mesh networks for both the arbitrary and random models as discussed in [6]. It is shown that, for a network of n nodes, the network capacity is $\Theta\left(\sqrt{\frac{n}{log(n)}}\right)$ when the ratio of channels to interface is bounded by $O(log(n))$ for a random topology i.e. there is no capacity loss in multi-radio multi-channel networks when scaled from a single channel. This result is further validated by analysis in [7], in which the authors drive the upper bounds on capacity that can be achieved. These studies clearly show that multiple radio usage with multiple channels can effectively increase capacity.

The capacity in a single radio single channel case is limited first by it being half duplex. Adding another radio can make it full duplex but than too it cannot be used simultaneously because of interference of using the same channel. Also, as shown in [5], the experimental throughput decay for a single channel multi-hop ad-hoc networks for a node behaves as $1/(n^{(1.68)})$, n being the number of nodes. If all the nodes interfere with each other (if sensing range of all nodes is greater than the network diameter) then we have a throughput bounded by $O(1/n)$. Similarly, for a chain topology, the throughput observed is 1/8 of the total throughput [9].

If we have multiple non overlapping channels available, with a single radio, then a node may switch channels dynamically to increase capacity [14]. A node still cannot work in duplex mode but switching channels can provide simultaneous transmission on orthogonal channels in disjoint nodes. The problem this entails is that though the throughput is enhanced by this technique, the channel switching cost is present. The switching delay for commodity hardware ranges from a few milliseconds to a few hundred micro seconds. This leads to an increase in the end to end delay. Also, channel switching is difficult to implement as nodes using different channels cannot communicate and require synchronization between them.

Multi-radio multi-channels approach avoids all the above drawbacks present in single radio approach. A node can now operate in full duplex mode, sending and receiving data on non overlapping channels. Also, with static channel binding, there is no channel switching cost. Using multiple channels much greater than the number of radios further increases throughput. But to realize this gain we need a channel assignment that makes this possible. This then leads us to the problem of efficient channel assignment to increase capacity.

In this paper, we make a comparison of different static, multi-radio channel assignment schemes that do not take traffic into account. Not taking traffic into account translates to all links having the same priority and having the same chance of being used so that the fairness is maintained. Normally, in the literature, if we consider the wireless mesh network, we are considering a network of access points that

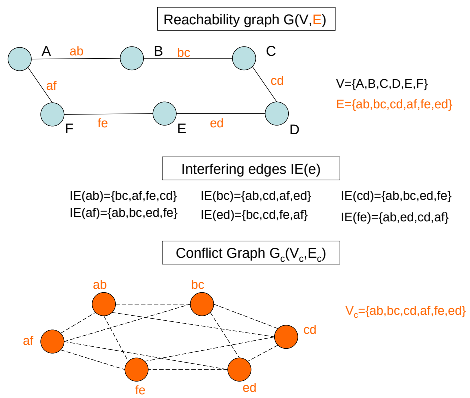


Figure 1: Reachability Graph, Interfering Edges and Conflict Graph

are fixed with wireless connections and having multiple radios. In such a case, the traffic usually moves towards the access points that also serves as a gateway to Internet and hence traffic pattern is defined. But in this comparison, we are considering strategies which do not consider traffic or routing while assigning channels since our aim is to compare strategies not for access points but for nodes in an ad-hoc network.

In this paper, we compare different centralized algorithms so as to provide a benchmark of what can be achieved. The algorithms studied are such that they can be easily decentralized. As explained in section 2, the channel assignment problem is NP complete. Hence our comparison provides a benchmark of what can realistically be achieved when using a distributed approach. A multi-radio network is considered since the prices of NIC's have dropped so much that it has become feasible to add more than one NIC's in devices thus providing extra capacity to meet the requirements for the ever increasing bandwidth. But for this to happen, we need algorithms that can intelligently select channels so that the capacity increase is realized.

1.1 Contribution and Paper Organization

This paper, first of all, introduces a new algorithm, namely MCAIR algorithm, for channel assignment and at the same time proposes a variant of random channel assignment to maintain maximum connectivity. Additionally it provides an extensive comparison of the existing algorithms for different random topologies. We also simulate the grid topology which corresponds to mesh networks and provide results. We also introduce some new metrics which help in better judging the channel assignment than just interference.

This paper is organized as follows. In the next section, the graph based mathematical model and notations used in this paper are described. We then introduce the different algorithms used in this paper in Section 3. Section 4 gives the simulation results with the existing metrics. New metrics and results based on them are then given in Section 5. We conclude in Section 6.

2. MATHEMATICAL BACKGROUND

Reachability Graph.

Consider an undirected graph $G(V, E)$ that models a communication network. The vertices/nodes set V consists of the nodes of the network, which may have multiple radios (not necessarily same), while the edges/links set E are the communication links in the network. A link e between a pair of nodes (v_i, v_j) ; where $v_i, v_j \in V$ exists if they are within the radio range of each other and use the same channel. The graph G described above is called the *Reachability graph*.

The Channel Assignment(CA) problem is now to assign channels to the radio interfaces present at each vertex while maintaining all the links as specified in the reachability graph. This approach is called "topology preserving". Mathematically, this can be posed as either assigning channels to vertices or to edges respecting the constraints of connectivity and of limited number of radios. The former is known as vertex formulation while the latter is called edge formulation. Both of these formulations are equivalent and interchangeable. It can also be seen that there exists a feasible solution; i.e. assigning same channel to all nodes. The goal of a channel assignment algorithm is now to assign channels while respecting the constraints so as to optimize a criteria like minimization of interference for increasing capacity.

Interfering Edges.

To achieve the above stated goal, we need to include the interference in our model of the network. For this we first introduces the concept of Interfering edges. Interfering edges for an edge e, denoted by $IE(e)$, can be defined as the set of all edges which use the same channel as e and which cannot be simultaneously active with e. Since all the edges are competing for the same resource(channel), hence the aim of a CA algorithm is to minimize this set for all edges e thereby increasing capacity.

Conflict Graph.

Based on the notion of interfering edges, we now introduce the concept of conflict graphs. A conflict graph $G_c(V_c, E_c)$ consist of set of vertices V_c and edges E_c. The set V_c has a one to one relation with the edge set E of the reachability graph; i.e. for each edge $e \in E$, there is a $v_c \in V_c$. As for the edge set E_c of the conflict graph, there exists an edge between two conflict graph vertices v_{ci} and v_{cj} if and only if the corresponding edges e_i and e_j of the reachability graph, are in $IE(e)$ set of each other. Simply put, if two edges interfere in the reachability graph, then there is an edge between them in the conflict graph.

The conflict graph can now be used to represent any interference model. For instance we can say that two edges interfere if they use the same channel and are less than two hops away. If any other model based on signal power is used then that can also be easily incorporated by just defining the conditions of interference. Total interference can now be described as the number of links in the conflict graph i.e. the cardinality of E_c.

The above concepts of reachability graph, interfering edges and conflict graph are illustrated in figure 1. For a graph $G = (V, E)$, we find the IE for all the links and then construct the conflict graph $G_c = (V_c, E_c)$. Note that here we consider that links interfere if they are two or less hops away.

Channel Assignment(CA) problem.

The CA problem in terms of conflict graph can now be thought of assigning channels to V_c such that we minimize the sum of degrees of $v_c \in V_c$ (i.e. the overall network interference) while respecting the radio constraints. Thus, the CA problem is to compute a function $f : V_c \rightarrow K$; where K are the available channels, to minimize the overall network interference while satisfying the radio constraint.

The channel assignment problem is NP-hard as shown in [12]. Authors in [15] have also equated it to the Max K-cut problem which in itself is NP-hard. Hence no simple solution exists and various heuristics have been used by different authors based on different constraints.

3. ALGORITHMS

Below we present the algorithms that we have considered in this paper. These are heuristics that are most efficient/appropriate from those proposed in the literature for an ad hoc network.

3.1 Random Channel Assignment

Random Channel Assignment, as it's names implies, is alloting of channels randomly without taking any criteria into account. But if such an assignment is done from all the available channels K, then we might obtain a topology that is not connected at all. So to obtain a somewhat connected topology, we have modified the random channel assignment algorithm. Now, we iterate through all edges $e \in E$ and assign channels in the following way.

- If both the nodes/vertices v_i and v_j forming a link e have a free radio, then we randomly assign a channel to it from all the possible channels available(from K). If the nodes are already using the same channel to communicate with some other nodes, then we need not commit another radio for the same channel but we use the same radio for communication on this link also.

- If only one node v_i forming the link has a free radio then the link is formed by selecting randomly a channel from the channels used by the other node v_j(having no free radio). If the selected channel is already in use by a radio at v_i, then we need not assign a free radio to the same channel.

- If none of the node have a free radio then the edge e is assigned a randomly selected channel from the available common radios of the two nodes.

- In case none of the above conditions hold true i.e. both the nodes have no free radio and have no common channel, then the edge e remains unassigned.

As can be seen from the above description, its possible that all the edges e in the reachability graph G may not be covered after the channel assignment. Thus the random channel assignment is not generally topology preserving. It becomes topology preserving when for n radios the number of channels are less than or equal to $2n - 1$. For example if there are 3 radios then the maximum number of channels that can be used so that the topology is preserved is 5 and so on. It is simple to verify the above assertion. If for example we continue with the example of 3 radios, than if each node has 3 radios and the maximum number of channels is 5, then two neighboring nodes always have a common radio; thus preserving the topology. It is also seen from this discussion that greater the difference between the number of channels and radios; greater is the chance of topology partitions.

3.2 Greedy Algorithm

In this algorithm, presented in [15], channels are greedily assigned based on the criteria that each assignment tries to minimizes the overall network interference. But this is not simply a one pass algorithm as we can select the same link multiple times for optimization.

We start by first assigning all links to the same channel (single channel case). Thus we have an initial assignment with maximum interference and we try to optimize this assignment by changing one link at a time with respect to constraints. Thus this algorithm is topology preserving since we can only modify a link's channel assignment if it satisfies the constraints and if we can find a new channel that reduces the overall network interference.

On each iteration of the algorithm, we randomly select a link to be improved (since link is selected randomly, hence it may be selected multiple times for improvement). Then we try all possible combinations with this link to find the one that minimizes the network interference. The number of combinations are determined based on the constraints of the nodes sharing the link. Like in the random algorithm, if both the nodes have free radio, then the channel with the leasts interference is selected. Else if only a node has a free radio, then we select the channels from those that are in use by node having no free radio that minimizes interference. Here note that two nodes sharing a link always have one common channel since we have started with a connected topology unlike in random algorithm.

We continue such iterations until we do not have any interference improvement for at least $|V|$ number of iterations. Whenever we have improvement we reinitialize the counter.

3.3 Clica Algorithm

Clica [10] is a single pass greedy algorithm, that assigns channels to nodes in order of their priorities. The priorities are determined by the number of free radios at a node. If we assign a radio a channel, then the free radios at node decrease by one. Thus clica must dynamically adjust priorities as it assigns channels. This makes clica recursive, i.e. if on assigning a link a channel, thus fixing one radio at a node, we find that our neighboring node has no more free radios and has thus loss flexibility, then we assign channels for that node immediately thus changing the priority dynamically.

The functionality of Clica algorithm can be divided into three parts. We first find all the neighbors of a node and then find the common channels if any between our node and its neighbors. If such a channel exists then it is assigned to the link. The second part looks at the *nodeset*, a set which contains nodes which have no free radios. It then tries to find all possible uncolored paths (link combinations to whom channels have not been assigned) which have only one radio unassigned to all nodes in *nodeset*. If such a path is found, then we assign a channel/color to the link and enter into recursion since the link assignment causes no free radios on the next hop. While doing so, we also add our node in the *nodeset*. The third part of the algorithm is concerned with assigning all uncolored links of a node not covered in first and second. For a link, we select a channel greedily based on the criteria. If all radios are already in use then it is selected from the channels being used else we select the channel from all available channels. The channel is then assigned to the link and the radio, if necessary. If this assignment results in having no more free radios at our neighbor, then we increase its priority and visit it by entering into recursion. In such a case, the node is included in *nodeset* if it has no free radios.

In our simulations, when we tried clica on networks with high node degrees and with channels much greater than radios(i.e we have channels that are equal or more than twice the number of radios), we found that sometimes clica blocks.

This may be a problem of our interpretation of the algorithm. In our case, this situation arises because with above condition, it is possible to have two connected nodes whose radios are assigned such that there is no common channel between them. Thus, clica in such case does not preserve the original topology. This problem arises due to multiple recursions. When radios of a node has been already assigned, it goes into recursion but all its links may not have been assigned. If this happens multiple times, then there might arise a situation in which two adjacent nodes have radios on different channels and hence the link between them cannot be allocated, thus breaking original topology preservation (the condition of finding all paths having a single free radio between the nodeset and current node is not sufficient to avoid this). Therefore, in the simulation results, clica is not shown.

3.4 Tabu Algorithm

In Tabu algorithm [15], we initialize by randomly assigning the channels to links. The channel for each link is selected randomly from the set of K available channels. This assignment is done without taking radio constraints into account. Then we try to improve the assignment in each iteration. For this, we prepare r new assignments; each being different from the original by one link channel assignment. These r assignments are then compared to find the one that reduces the interference the most. This assignment is then chosen as the next base case from which r different assignments are prepared. To escape the local minima, the assignment chosen between the r may have greater interference than the original base case. We terminate the algorithm when we cannot find any improvement in the best case (saved separately from the base case) for the $|E|$ number of iterations. To speed up things, we maintain a Tabu list of limited size in which we guard the solution selected from the r assignment in each iteration. Before calculating a new solution, we check if this solution is already present in the Tabu list. If it is, then we don't need to select the same solution again.

When the first phase terminates, we have a channel assignment that minimizes the interference. But this assignment does not respect the radio constraint that must be followed at each node. We now apply a procedure which makes this assignment respect the constraints of limited radios at each node. Here we first select a node with maximum violations, i.e. a node that uses the maximum number of channels and has the maximum connections. In case of a tie, we randomly select the node. Then we select two channels being used by the node and merge the two channels into one channel. The channels are selected such that their merging causes the minimum of interference increase. Since the nodes may be using the same channel to connect with other nodes as well, which may be connected to other nodes and so on, hence the merging of channels is propagated to all connected nodes in the network using the channels. Thus a merge iteration, decreases the conflict for all nodes in the network and does not cause any conflict increase at any node. We continue such merge operations until no node in the assignment violates the radio constraints.

From the above description it is clear that the Tabu algorithm unlike random or clica is not a one pass algorithm. Also, the computational time of tabu algorithm is large as we have to prepare r solutions (we have taken $r = V/2$; where V is the number of vertices in the reachability graph) and calculate the total interference of each solution, from which one is selected in each iteration. Also, we have a minimum of $|E| = V_c$ iterations($|E|$=number of links in the network).

3.5 Merge Based Channel Assignment for Interference Reduction(MCAIR)

In Greedy algorithm, all links are assigned the same channel initially (worst possible assignment). Then it tries to improve this assignment in each iteration. On the other hand, Tabu assigns channel randomly in the begining without applying the radio constraints. It then tries to improve its assignment and then finally applies the constraints. In both cases the initial assignment is far from optimal. The idea of mcair algorithm is to start with an optimal assignment. This assignment may violate the constraints of the limited number of radios and also of the limited number of channels. The merge procedure is then applied to make the algorithm converge to a feasible solution(one that does not violate the constraints).

This algorithm is based on the Brooks theorem, which states that the vertex coloring of a graph with degree d is possible with d or $d+1$ colors ($d+1$ colors are required for odd cycle graphs and complete graphs) [3]. Vertex coloring is assigning colors to each vertex of a graph such that no edge connects two identically colored vertices. The vertices of a conflict graph (V_c) gives us the links of reachability graph and E_c gives us links which interfere with each other. Hence what we do is find the degree of conflict graph and then without taking the constraints of channels and radios into account, try to vertex color the conflict graph. This gives us an initial assignment with zero interference.

To achieve this, we select a vertice v_c of the conflict graph and then find all the colors used by its neighbors. We assign the first unused color we find to this vertice. We iterate through the above process until all vertices are colored. This gives us a zero interference assignment of graph without considering the constraints. To make the above solution respect the constraints, we apply two merge procedures. First to reduce the number of channels used to the number of elements in K and the second so that each node respects the radio constraint.

The first merge procedure consists of finding all possible combinations of the channels and merging the two channels whose merge causes minimum interference increase. This process is repeated until the number of colors used is equal to the number of channels. Next we apply the radio constraint. For this we use the same merge procedure as described in Tabu until all vertices respect the radio constraint.

4. SIMULATION RESULTS

The algorithm comparison was done by implementing the different algorithms in MATLAB. For comparison, we have considered two types of graphs. Random graphs where n nodes are uniformly distributed in a square of dimension L. In these graphs, a link exists between two nodes if their distance is less than their radio range R. For simulations, we have used $n = 25, 50$ and 75 with $L = 1000m$ and $R = 250m$. The different values of n allow us to explore the impact of different density of nodes(average degree of a node is 3.8,7.5 and 11.5 for 25,50 and 75 nodes respectively). The other graph we considered is the classical grid where interior nodes have degree 4, nodes on the sides having degree 3 and corner nodes having degree 2.

As for calculating the conflict graph, we assume that the interfering edges of a link is the set of links within two hops distance if they are on the same channel. This is also the

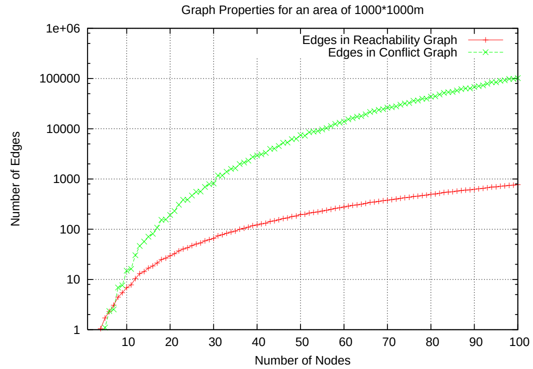


Figure 2: Number of Edges in Graph and Conflict Graph

case when RTS/CTS mechanism is used for 802.11 based networks.

Figure 2 shows the number of edges in the random graph and its conflict graphs when graph size varies from 1 to 100 nodes. Take note that we use the logarithmic scale. As can be seen from the figure, the increase in the edges of conflict graph is exponential. This has implications that if an algorithm does not guarantee complete topology preservation, then its comparison with other algorithms that does the same is of no use. So the Random algorithm cannot really be compared with the other algorithms. It has been given here in some figures, just to make our point.

The metric usually used for calculating improvement is the fractional interference. It is defined as the ratio of the number of edges in conflict graph after link channel assignment to the number when using a single channel. Thus it gives the reduction in interference achieved due to the use of multiple radios and multiple channels.

4.1 Results for Random Graph

Figure 3 shows the comparison with 50 nodes when using 2, 3 and 4 radios with respect to available channels (one figure per algorithm). As can be clearly seen increasing even a single radio has a significant impact. Using only 2 radios per node reduces the interference by almost half (fractional interference is below 0.6 for all the algorithms). Increasing the number of radios further results in decreasing the fractional interference further though the rate of decrease is more gradual than the rapid decrease seen earlier.

Another important thing that is observed for greedy algorithm is that increasing the number of channels above a certain threshold causes increase in the interference. This counter intuitive result can be explained if we take a look at how algorithm works. This happens as all nodes initially use the same channel and then we try to reduce the interference by changing channels. Having more channels at our disposal causes the interference to be reduced for a few links but the radio constraint skews the distribution of links towards the initial channel. Hence more channels causes more links to use the same initial channel increasing overall interference.

Figure 4 shows the comparison of algorithms for 2,3 and 4 radios w.r.t the number of channels. The number of nodes is 25. When there are only 2 radios at each node, it is seen that all the algorithms perform more or less the same as shown is figure 4(a) except the random algorithm. The performance of random algorithm does not count as we are not sure if it even preserves the network connectivity. When the number of radios are high, and we use the same 25 nodes, as in figure 4(c), the topology is more or less preserved for the random algorithm also. Here, in figure 4(c), it can be seen that its performance is much worse in comparison. It is seen from the figures that greedy algorithm performs the best when its channels are bounded (3, 5 and 8 channels for 2,3 and 4 radios respectively).

When the same comparison is done for 75 nodes, as shown in figure 5, we find that greedy algorithm is better for the above said points but when the number of channels are increased, its performance deteriorates. Also, it is remarked that the tabu and mcair algorithms do not vary greatly with the number of channels. The reason being that the radio constraint has far greater impact than the number of channels. When the merge is applied to a violating node in both algorithms to respect the radio constraint, its effect is propagated. Since channels are evenly distributed, this causes the effect to be propagated throughout the network. Hence the variation with the number of channels is limited for the two algorithms.

Also, note the scale of the fractional interference between figures 3, 4 and 5. It can be seen that with increasing number of nodes, the fractional interference decrease diminishes.

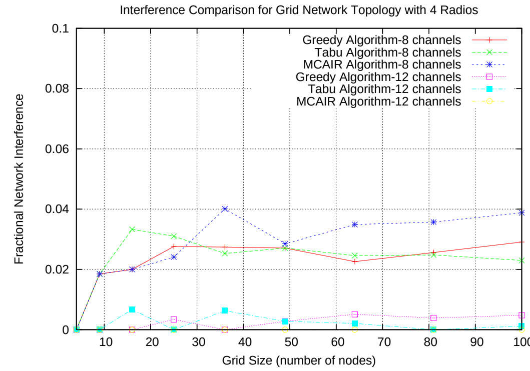


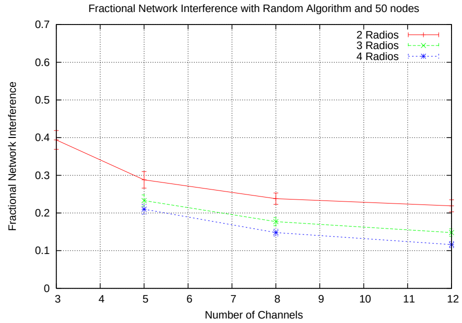
Figure 6: Grid Fractional Interference

4.2 Results for Grid

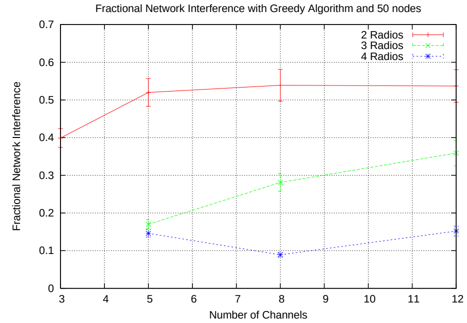
The pecking order of the results obtained for the grid are the same as random graph and are thus not shown here. However, we show figure 6, the comparison for 4 radios with 8 and 12 channels for different grid sizes. We first observe that the gain with the use of several channels is huge (maximum fractional interference of 0.04 for grid compared to 0.2 for random graphs). Also, with mcair algorithm, with 4 radio and 12 channels, we find that there is no interference for any grid size. This is explained by the fact that since the maximum degree of node is 4 in our grid for any node, hence with 12 channels, mcair algorithm is able to assign channels such that the 22 edges (maximum Interfering Edge set) that interfere have different channels (channels are repeated but in such a way that channel reuse is done only for those edges that are more than two hops thus causing no interference).

5. NEW METRICS

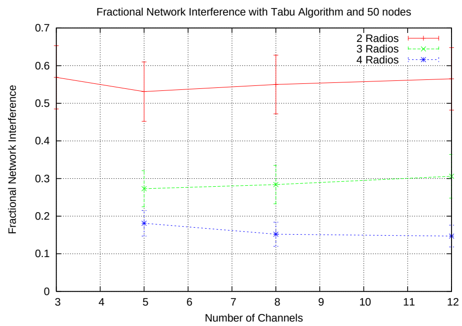
Since the aim of using multiple radios and channels is to increase capacity, hence using only fractional network interference to measure capacity increase is not sufficient. It suffers from the problem that since the fractional network interference is the sum of interference observed on all channels divided by interference observed when using a single channel, hence it does not tell us anything about the interference distribution in individual channels. We now introduce new metrics that try to capture this.



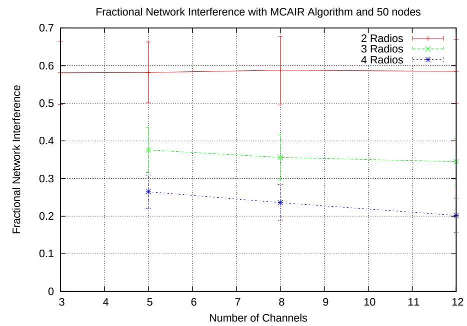
(a) Random Algorithm-50 nodes



(b) Greedy Algorithm-50 nodes

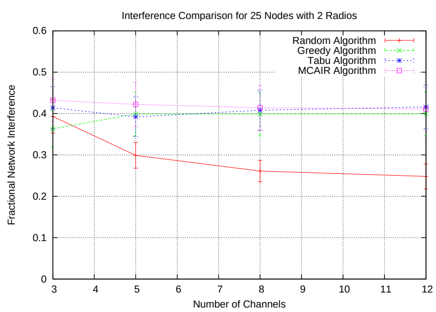


(c) Tabu Algorithm-50 nodes

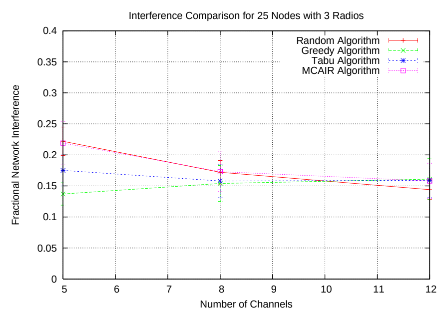


(d) MCAIR Algorithm-50 nodes

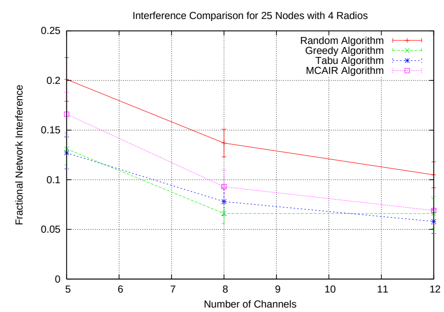
Figure 3: Algorithm Comparison for a topology with 50 nodes



(a) Algorithm Comparison-25 nodes with 2 Radios

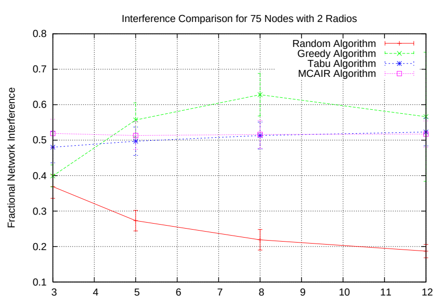


(b) Algorithm Comparison-25 nodes with 3 Radios

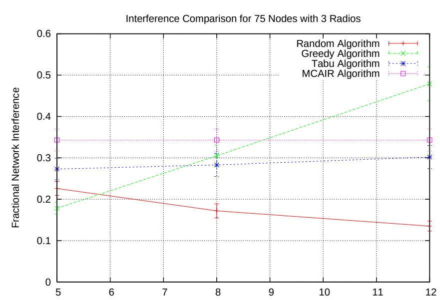


(c) Algorithm Comparison-25 nodes with 4 Radios

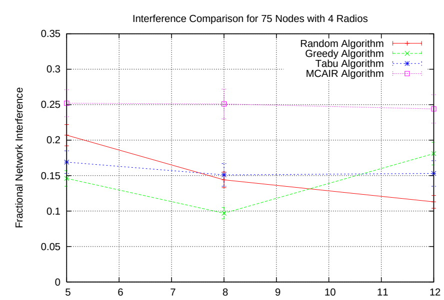
Figure 4: Comparison of Algorithms - Graph with 25 nodes



(a) Algorithm Comparison-75 nodes with 2 Radios



(b) Algorithm Comparison-75 nodes with 3 Radios



(c) Algorithm Comparison-75 nodes with 4 Radios

Figure 5: Comparison of Algorithms - Graph with 75 nodes

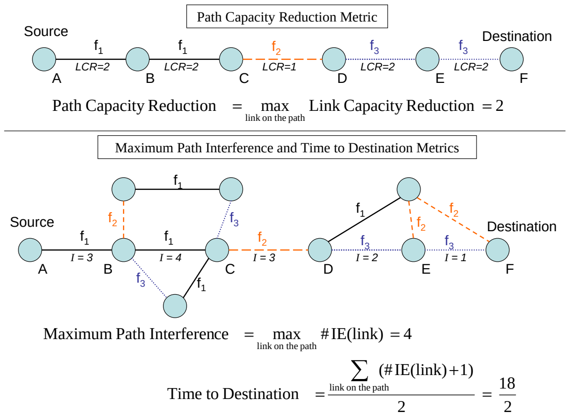


Figure 7: Metrics Explanation

- **Path Capacity Reduction**. To calculate this metric, a path between any two arbitrary nodes is found after the channel assignment. Then for each link in the path, we calculate the number of interfering links (including itself) which are on the path. This is called the link capacity reduction for each link. We then consider the maximum value of link capacity reduction over the links for the path. Since the path is always a chain, hence the maximum value is limited to 5 (corresponding to a portion along the path where the same channel is used on 5 consecutive links). This can be though of as the capacity division factor i.e. the capacity is divided between the five links leaving capacity for a link equal to $c/5$ if c is the total bandwidth as only a single link can be active at an instant. Also, this is the bottleneck or limiting link which determines the overall throughput of the path.

  Figure 7 gives an example calculation of this metric. Consider a path from A to F. We calculate the link capacity reduction of each link along the path. This is calculated by simply regarding the predecessors and successors until two hops and finding the number of links that use this channel. For link $B \leftrightarrow C$, it will look up channels used by link $A \leftrightarrow B$(predecessor) and links $C \leftrightarrow D$ and $D \leftrightarrow E$ (successors). Since $A \leftrightarrow B$ uses the same channel hence link capacity reduction for link $B \leftrightarrow C$ is 2. The path capacity reduction for this path, when calculated from $A \leftrightarrow F$, also equals 2.

- **Maximum Path interference**. Unlike the previous metric, this metric also takes into account the interference from other links not on the path hence its value is not bounded. This gives us not only the bottleneck link along the path but also provides an idea about the delay that a packet may suffer on average (a chain is as strong as its weakest link). Using the notation introduced in Section 2, it can be formally defined as $max|IE(e)|$, where $e \in Path$.

  In figure 7, link $B \leftrightarrow C$ has the maximum path interference value as channel f_1 is being used by four other links in its two hop neighborhood leading to a metric of 4 for the path from A to F.

- **Average Time to Destination**. This metric is similar to the previous metric, namely maximum path interference, except here, instead of taking the maximum, we sum the interference. It is formally defined as $\sum_e (|IE(e)| + 1)/2$, where $e \in Path$. It provides an idea of the total delay that a packet may suffer. Let t be the time needed to transmit a message from a node to its neighbor when there is no interference from other links. When there is interference, the time to transmit increases as the number of links contending for the channel increases. If there are two contending links then we say that the time to transmit a packet on average is $3t/2$ as it can send in time t if it gains immediate access to channel; otherwise it requires time $2t$.

5.1 Metrics Result

We now compare the algorithms using the above described metrics.

Random Graph results.

For this we have used 50 nodes topology with 8 available channels. The comparison results are given in figure 8. The results here show that greedy may not be the best choice even when it is bounded. Like for path capacity reduction, with 4 radios and 8 channels, we find that tabu algorithm has value of around 1.2 as compared to around 1.5 for greedy algorithm. Overall, we find that for random graphs, when all metrics are taking into account, tabu gives better results than greedy.

Grid results.

Figure 9 uses the above metrics to compare the algorithms for a 10*10 node grid with 8 channels. The three algorithms offer similar results when considering path capacity reduction (figure 9(a)). For both maximum path interference and average time to destination (figures 9(b) and 9(c)), mcair offers better results for 2 and 4 radios. Mcair is clearly better when using 4 radios as it reduces interference to zero.

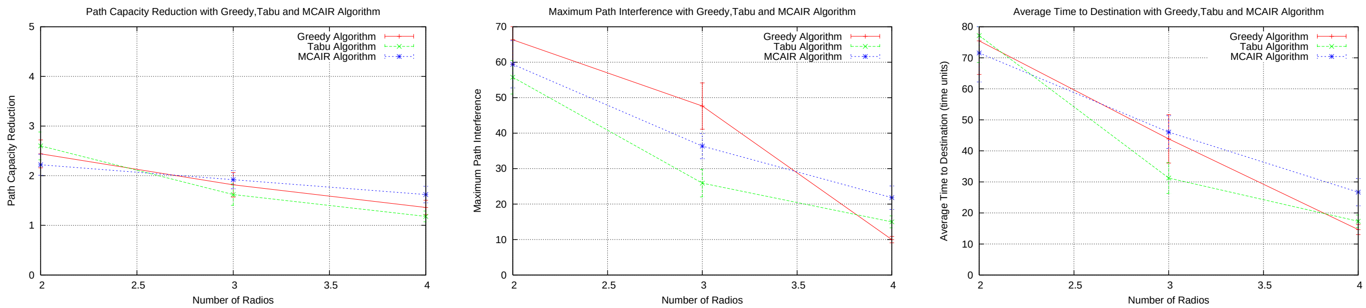
6. CONCLUSIONS

In this paper, we have compared different static centralized channel assignment algorithms. The comparison shows that we have different results based on the kind of graph. The greedy algorithm is generally better when we consider the total interference (number of links in the conflict graph). But this is true only if the number of channels are bounded based on the number of radios. Indeed if the number of available channels is large, then interference increase is observed if the number of radios remain constant. This strange counter intuitive behaviour has not been reported in earlier studies. For grid networks, with 4 radios, mcair algorithm, proposed in this paper, is best as it completely eliminates interference. This is an important results as in real world scenarios, degree of nodes for a mesh network is small and they are often equipped with 4 radios. If we consider other metrics, like capacity of a path for instance, then the hierarchy between the algorithms is changed. Tabu algorithm in this case comes out to be better than others.

7. REFERENCES

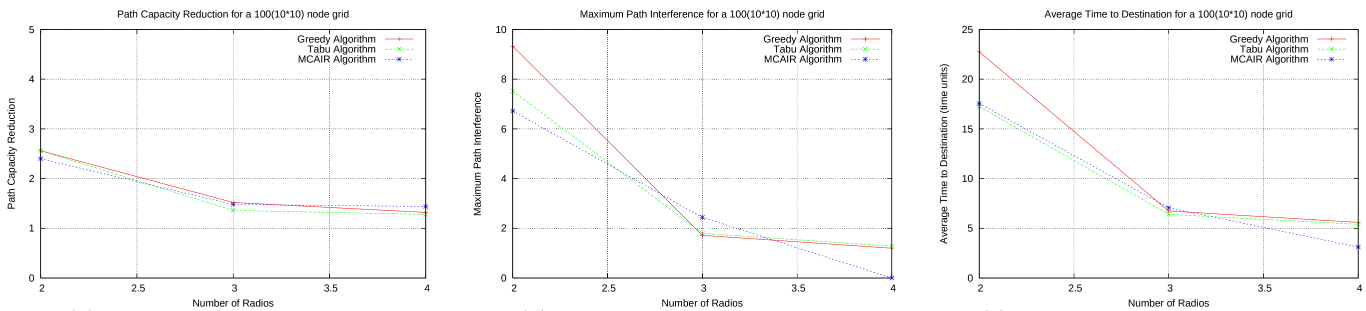
[1] IEEE standard for information technology - specific requirements part 11. Technical report, 2005.

[2] M. Alicherry, R. Bhatia, and L. E. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 58–72, New York, NY, USA, 2005. ACM.



(a) Path Capacity Reduction (b) Maximum Path interference (c) Average Time to Destination

Figure 8: Results with new metrics for random graphs



(a) Path Capacity Reduction (b) Maximum Path interference (c) Average Time to Destination

Figure 9: Results with new metrics for grids

[3] R. L. Brooks. On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37(02):194–197, 1941.

[4] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy. Optimization models for fixed channel assignment in wireless mesh networks with multiple radios. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, pages 481–490, San Diego, CA,, June 2007.

[5] P. Gupta, R. Gray, and P. R. Kumar. An experimental scaling law for ad hoc networks, 2001.

[6] P. Gupta, S. Member, and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46:388–404, 2000.

[7] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 73–87, New York, NY, USA, 2005. ACM.

[8] P. Kyasanur, , and N. H. Vaidya. Capacity of multichannel wireless networks under the protocol model. *IEEE/ACM Transactions on Networking*.

[9] J. Li, C. Blake, D. S. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69, New York, NY, USA, 2001. ACM Press.

[10] M. K. Marina and S. R. Das. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. In *Broadband Networks, 2005 2nd International Conference on*, pages 381–390, Oct. 2005.

[11] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted coloring based channel assignment for wlans. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(3):19–31, 2005.

[12] A. Raniwala, K. Gopalan, and T.-c. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004.

[13] H. Skalli, S. Ghosh, S. K. Das, L. Lenzini, and M. Conti. Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions. In *Communications Magazine, IEEE*, volume 45, pages 86–95, Toronto, Ont., Canada,, Nov. 2007.

[14] J. So and N. H. Vaidya. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 222–233, New York, NY, USA, 2004. ACM.

[15] A. P. Subramanian, H. Gupta, and S. R. Das. Minimum interference channel assignment in multi-radio wireless mesh networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, pages 481–490, San Diego, CA,, June 2007.

[16] J. Tang, G. Xue, and W. Zhang. End-to-end rate allocation in multi-radio wireless mesh networks: cross-layer schemes. In *QShine '06: Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks*, page 5, New York, NY, USA, 2006. ACM.

[17] T. D. H. Timo Vanhatupa, Marko Hännikäinen. Optimization of mesh wlan channel assignment with a configurable genetic algorithm. WiMeshNets, 2006.

[18] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-L. Sheu. A new multi-channel MAC protocol with on-demand channel assignmentfor multi-hop mobile ad hoc networks. In *Parallel Architectures, Algorithms and Networks, 2000. I-SPAN 2000. Proceedings. International Symposium on*, pages 232–237, Dallas, TX, USA, 2000.