Channel Assignment for Wireless Meshes with Tree Topology

Dragoș Niculescu
ETTI, University Politehnica of Bucharest,
Romania

Sudeept Bhatnagar

Samrat Ganguly
NEC Corporation of America,
USA

***Abstract*— The capacity of wireless mesh networks can be enhanced with judicious channel assignment. This paper deals with one particular type of mesh network topology – the tree topology. The unique characteristics of this topology is that all the traffic to/from the mesh nodes goes through the root. This enables design of an efficient channel allocation algorithm that utilizes the intrinsic characteristics of the tree topology and the traffic pattern over this topology. We use the unique connection characteristics of the tree topology to create an auxiliary contention graph over which we execute our coloring algorithm. This mitigates the burden of ensuring connectivity that channel allocation algorithms for mesh have to consider and the algorithm can solely focus on the task of capacity maximization. Our algorithm has a low complexity of $O(N^2)$ for a mesh network with N nodes.**

I. INTRODUCTION

Wireless mesh networks are used to transport data over multiple wireless hops. The capacity of a wireless mesh network is affected by the degree of channel reuse among its wireless nodes. IEEE wireless standards have the provision to use multiple channels in the network and mesh nodes typically have multiple radio interfaces which can be configured to use distinct channels. An efficient channel allocation algorithm can exploit the channel diversity provided by the wireless standards to maximize the capacity of a wireless mesh network by tuning different interfaces to different channels.

Any channel allocation algorithm has the topology knowledge and optionally the traffic pattern information to aid in its decision making. This work deals with a specific case of wireless mesh networks where the routing structure is a tree. In this topology, all the mesh nodes act as the access points for clients and all the traffic flows through the root node. The root node acts as the gateway for all communications. Each mesh node forwards its traffic towards the root possibly over multiple hops and all the traffic meant for that mesh node follows the reverse path from the root to itself. We leverage these characteristics to design an efficient channel allocation algorithm specifically for tree-structured wireless mesh networks.

II. NETWORK MODEL & PROBLEM STATEMENT

We consider a wireless mesh network of N nodes. Each node has two interface cards to communicate on the mesh. The mesh nodes may also have a third interface card which is used to communicate with the clients. The third interface card uses a different carrier than the two used for communicating with the mesh nodes (for example 11a vs. 11b). The interference set I_k for node k is defined as the set of nodes which can interfere with k if they transmit on the same channel (using either of the interfaces). We consider the Boolean model of interference where a node either interferes or does not interfere with any other node. There is no notion of partial interference or the actual amount of interference (packet loss) that a node causes. The simplicity of this interference model is important to keep the interference information obtainable in a practical setting.

The routing structure for the network is a tree. One of the nodes is designated the root node and it acts as the gateway for all traffic whether it is destined to another mesh node or to a node outside the mesh (possibly on the Internet). All other nodes act as access points for the clients and route the traffic to/from the root. The routing structure imposed on the topology is a tree with the root node being the gateway for all traffic. Each of the other nodes uses its mesh access cards as upcard or downcard (Figure III). All traffic towards the root is sent on the upcard and all traffic from the root meant for one of the downstream mesh nodes is forwarded on the downcard. All traffic to/from the client uses the client access card. The root node uses both its mesh access cards as downcards since it does not have any parent in the mesh. This serves the purpose of increasing the mesh capacity since the root is likely to become the bottleneck in such a scenario. Using two downcards (on different channels) doubles the amount of traffic that the root can source.

A node k has t_k units of traffic to send towards the root. Since all traffic goes to/from the root, the traffic that a mesh node sends to its parent (towards the root) is the sum of traffic from all its children and the traffic generated by the clients. Similarly, all traffic it receives from its parent is the sum of traffic meant for its clients and its children. Thus, the total traffic at the nodes higher up in the tree is going to be higher than the total traffic at the nodes lower in the tree.

There are a fixed number of channels that we can use. For example, 802.11b has 3 orthogonal channels and 802.11a has 12. The exact number of channels that we can use is an input to our algorithm. The channels are considered orthogonal, i.e., each channel is essentially unaffected by the traffic on any other channel. We do not consider partially overlapping channels like those possible in 802.11b. Our goal is to assign channels to the mesh access cards of the mesh nodes. We do not assign channels to the downcards of the mesh nodes which do not have any children in the tree (essentially these cards are not used). We do not consider the channel allocation between the mesh node and its clients and focus on the mesh portion only. Furthermore, we are not concerned with the routing portion of the problem; in fact in our model where all traffic flows to/from the root, the routing structure is equivalent to the tree topology. However, we recognize that joint channel allocation and routing is an interesting problem to study.

III. CHANNEL ALLOCATION

Since all traffic flows through the root, we know that the number of cards that the root node can have is not more than the total number of channels available in the technology the mesh uses. Otherwise, more than one of the cards that the root hosts will be forced to communicate on the same channel thereby splitting the throughput and rendering the extra card useless. In this section, we describe our algorithm to assign channels on the tree topology. The algorithm utilizes the properties of the traffic on the tree to assign channels. The algorithm also uses the information regarding which pairs of mesh nodes are within each other's carrier sense range.

978-1-4244-6363-3/10/$26.00

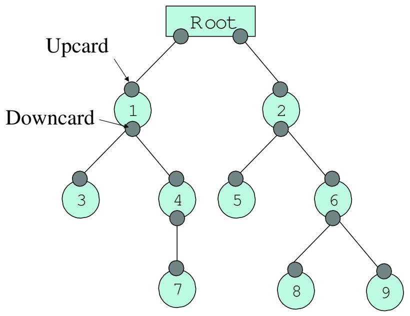

Fig. 1. A wireless mesh network with tree topology

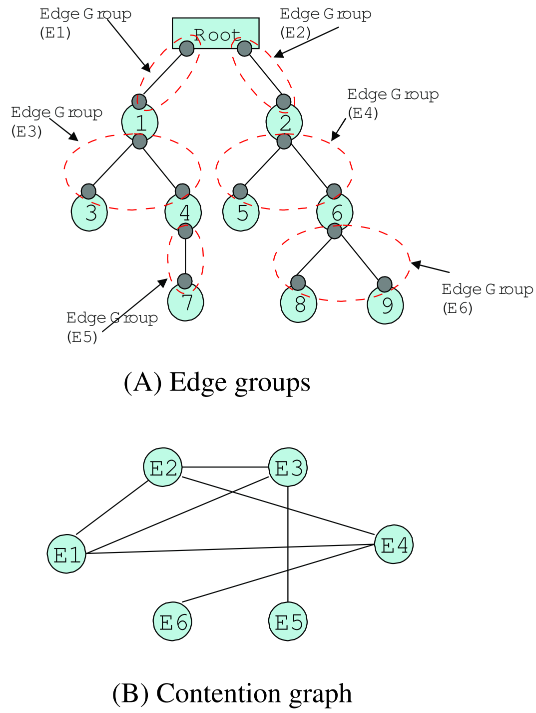

Fig. 2. Creation of contention graph with the following interface model: 1) Edge groups interfere if they have a common node; 2) Nodes 1 and 2 interfere

However, one of its limitations is that it does not use any information regarding the interference caused at a node by other nodes that are beyond its carrier sense range. However, our algorithm has been intentionally designed to use this simple information as interference modeling and measurement are not yet fully understood and require inordinate amount of time to build. Our algorithm is designed to use only the information which may be deduced in real scenarios.

A. Insights on the Tree Topology

We utilize the following information that is valid on a mesh network with a tree topology but not in a generic mesh:

- Since all traffic flows to/from the root, the total traffic at a node higher up in the tree (closer to root) is never less than the traffic at the nodes in its sub tree. In fact, the total traffic between a node and its parent is equal to the sum of the total traffic between itself and its children and the traffic that is sourced by the clients directly attached to it. This because the mesh node merely relays traffic from(to) the children node and its own clients, to(from) the parent node.
- It is better to avoid contention at the links that are higher up in the tree. This because if contention reduces the throughput of a link higher up in the tree, that reduction affects the traffic to/from all the nodes in the sub tree below the link. Hence if channel overlap is unavoidable, it is better to allow interference and higher contention in the lower level links which have less traffic than the higher level links anyway.

We use these insights in designing our algorithm. Before describing the algorithm, we first detail the creation of an auxiliary contention graph from the network topology to ease the task of channel assignment.

B. Contention Graph

One key requirement of the channel allocation algorithm is to allocate channels so that the connectivity of the network topology is maintained. In case of a tree topology, a channel allocation algorithm must never allocate separate channels to the up interface of a node and the down interface of its parent (else they would not be able to communicate).

To eliminate this possibility, we create an auxiliary graph over which our channel allocation algorithm executes. We create *edge groups* (Figure III-B) out of interfaces that must communicate on the same channel. Thus, all edges between a node and all its children constitute one edge group. The level of an edge group is the level (in the tree) of the parent node of its constituent edges. All edge groups corresponding to interfaces that are hosted on the root have a level 0. The level of the edge group e is denoted as h_e (height).

In the auxiliary graph, each edge group is treated as a vertex. We create an edge between two vertices A and B if any constituent interface of the edge group corresponding to A is in carrier sense range of any interface in the edge group corresponding to vertex B. The load on an edge group is the total traffic of all nodes in that group. Note that the traffic demand t_k at node k is part of the load of the edge group that contains its upcard and not the one that has its downcard. This because all traffic is destined for the root node in our case. The load on an edge group e is denoted as l_e . Furthermore, all edge groups that are neighbors of edge group e in the contention graph (are interferers) form its Interference set I_e . The channel allocation algorithm assigns channels to the vertices in the auxiliary graph. All interfaces corresponding to a vertex will be tuned to that channel. An example of a tree structured mesh and its corresponding contention graph is shown in the lower Figure III-B.

C. The Spread Algorithm

The algorithm aims at coloring the previously constructed contention graph with a given number of colors

(corresponding to the number of available channels in the network). It has the level and the load information of each vertex at its disposal. We refer to our solution as the *Spread Algorithm* since it primarily tries to spread the channels far apart depending on the load.

1)Vertex Traversal Order: The algorithm takes a single pass assigning colors to each vertex of the contention graph. Clearly, the order in which the vertices are visited plays an important part in the effectiveness of the channel allocation. If we visit an "important" vertex after assigning orthogonal channels to relatively unimportant vertices earlier, we may end up having to allocate already crowded channel to the important vertex leading to a loss in overall throughput.

In our case, the vertex traversal order is determined both by the level and the load of the vertex. We choose to visit a vertex at a lower level (higher up in the tree) early in the algorithm. If there are

G – set of all edge groups
l_e – load of edge group $e \in G$
h_e – level of edge group $e \in G$
I_e – interference set of edge group $e \in G$
$next(x)$ – element after x in LLPQ
C– set of channels

Fig. 3. Notations used in the algorithm

more than one such vertices, the priority is given to the vertex with a higher load. For efficient traversal, we build a priority queue using this logic. We call this queue the Level Load Priority Queue (LLPQ). The construction of LLPQ is shown at the top of algorithm III-C.

2) Channel Allocation: The channel allocation procedure gets the next edge group to assign channel by extracting the front element of the LLPQ. The impact of assigning each channel to this edge group is tested and it is assigned the channel whose throughput is expected to be affected the minimum with the new edge group operating on it. The question is how to deduce the impact of the edge group's traffic on the throughput of that channel.

Our rationale in determining the assignment utilizes the tree specific observation: If an edge group higher up in the throughput is affected, then the collective traffic from all nodes underneath that edge group is also affected. Thus, it is better to avoid hurting the higher level edge groups as much as possible. If there are multiple groups at the same level that are within the interference range of the group being assigned a channel, a better design choice is to use the channel of the less loaded group. These two insights serve as guidelines for our algorithm.

An important concept used in our algorithm is that of *virtual capacity*. The virtual capacity of the network is simply the maximum l_e over all edge groups. This serves as a guideline for maximum throughout in our tree based mesh. In an ideal scenario, whatever the absolute value of total load be, the edge group carrying the traffic equivalent of the virtual capacity is going to carry the maximum traffic. For example, if all nodes source unit traffic, then one of the two edge groups at the root is going to be the one determining the virtual capacity and also carrying the maximum traffic. Thusif that edge group is assigned a distinct channel, any other channel can be thought of as having a similar capacity (irrespective of the actual value of the traffic). This because of the implicit relationship between virtual capacity and the maximum possible traffic at any one interface.

The algorithm proceeds by extracting the first element (say E) of the LLPQ. It then checks the assigned channels for the neighbors of E in the contention graph. If there exists a channel that has not been assigned to any of its neighbors, then E is assigned that channel. If not, if there exists a channel whose load is low enough that even after reusing it for E, its total load remains below the virtual capacity, we assign it to E. If no such channel exist, i.e., for all channels there is a neighbor of E which is using it or if its total load will go beyond the virtual capacity, we find the channel for which the highest using edge group (among E's neighbors) has the highest level in the network. For example, if Channel 1 is being used by an edge group at level 2 and channel 2 is being used by an edge group at level 1 (both edge groups being neighbors of E in the contention graph), then we assign channel 1 to E. This rationale ensures that the channel used by E would try to minimize the hurt on the nodes higher up in the tree. If for two channels there highest level edge groups are at the same level, the channel having lower load is preferred for E. The

Algorithm 1 Functions *Used_Capacity* and *Lowest_Level* used in the algorithm

```
Used Capacity(c, e)
total_load ←0
foreach x ∈ I_e
    if assigned_channel(x) == c
        total_load ← total_load + l_x
    endif
endfor
return total_load

Lowest Level(c, e)
clevel ← ∞
foreach x ∈ I_e
    if (assigned_channel(x) == c)&&(h_x < clevel))
        clevel ←x
    endif
endfor
return clevel
```

Algorithm 2 Algorithm to assign channels to edge groups

```
Step 1: Create Priority Queue(G, l_e, h_e)
Initialize empty LLPQ
foreach e ∈ G
    x ← {y ∈ LLPQ|((h_y < h_e)&&(h_next(y) ≥ h_e))}
    if(x ≠ φ)&&(h_x == h_e)
        while ((l_x ≥ l_e)&&(h_next(x) == h_e))
            x ← next(x)
        endwhile
        insert e after x in LLPQ
    endif
endfor

Step 2: Assign_Channels
Virtual_Capacity ←max_{e∈G} l_e
while LLPQ is not empty
    e ← extract_first_element(LLPQ)
    if ∃ unused channel c ∈ C
        assign channel c to e
        continue
    endif
    if (∃c ∈ C) || (l_e + Used_Capacity(c,e) ≤ Virtual_Capacity)
        assign channel c to e
        continue
    endif
    chan ← (-1)
    clevel ← ∞
    cload ←∞
    foreach c ∈ C
        t ←Lowest_Level(c, e)
        u ←Used_Capacity(c, e)
        if((t < clevel) || ((t == clevel) && (cload > c)))
            chan ← c
            cload ← u
            clevel ← t
        endif
    endfor
    assign channel chan to e
endwhile
```

algorithm is shown as algorithm III-C. The algorithm's running time is $O(N^2)$ which is the time required to construct the contention graph. The channel assignment operation (assuming fixed number of channels) require less time since it primarily consists of scanning the neighborhood of each node to find the load on a given channel in its vicinity.

IV. RELATED WORK

Researchers have studied the benefits of using multiple channels and multiple radios in a wireless mesh network: authors in [1] proposed a new MAC layer to support multiple channels. Although a new MAC could utilize multiple channels more efficiently, but it would require modification to the existing 802.11 MAC. Our work is meant for use in the existing 802.11 MAC. Authors in [2], [1], [3] provide mechanisms for using multiple channels using a single interface. The protocol proposed in [2] does not require synchronization and can work with existing 802.11 MAC. Each node switches channels according to a pseudo random sequence, and it is guaranteed that the channels of any two

nodes overlap periodically. However, switching might also introduce delays. In this work, we consider multiple interfaces. [1] proposed the use of single interface to switch channels for load balancing. [3] proposed a protocol where each node with packet to transmit has to switch the channel of the receiver before transmitting data. Modeling the capacity of multiple channels and interfaces and understanding the benefits was studied by authors in [4], [5], [6], [7]. Reference [5] provides capacity model to understand the impact of the ratio between number of radios and channels on the system performance in the asymptotic case. [4] provides capacity model for multiple channels, using which feasibility of a rate matrix can be verified. Authors in [6] provide ILP formulation for throughput optimization in mesh network. They study the impact of interfaces and channels on the overall throughput. Contributions [8], [9] provide solutions to the joint channel assignment and routing problem. In [10], authors also proposed a framework for centralized channel assignment scheme for maximizing the throughput. In [11], authors proposed an adaptive channel assignment based on the congestion on a given link. [12] propose a link layer solution for striping data over multiple interfaces. [13] proposed a metric WCETT, which is suitable for mesh network with multiple channels. The network considers the channel interference and bandwidth apart from ETX measure. [14] considers the problem of creating a survival topology in a multichannel scenario and proposes a bandwidth aware routing algorithm. Also [15] proposes the use of partially overlapped channels to maximize benefits.

V. CONCLUSION

We studied the problem of channel allocation for wire less mesh networks with a tree topology. Specifically, in our scenario, each mesh node has two interface cards – the upcard to communicate with the parent and the downcard to communicate with the children. All traffic is routed through the root node of the system. We designed a novel algorithm that utilizes the knowledge of the topology and the knowledge of the traffic pattern if available, to assign channels to all cards.

Acknowledgment: This work was supported in part by the CNCSIS grant PN2 Resurse Umane 11/01.07.2009 and by FP7 grant SMART-net 223937. Parts of this work were performed while authors were employed by NEC Laboratories America, Princeton NJ, USA.

REFERENCES

[1] J. So and N. Vaidya. Multi channel mac for ad hoc net works: handling multi channel hidden terminals using a single transceiver. In *ACM Mobihoc*, 2004.

[2] Paramvir Bahl, Ranveer Chandra, and John Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 216–230, New York, NY, USA, 2004. ACM.

[3] N. Shacham and P. King. Architectures and performance of multichannel multihop packet radio networks. In *IEEE Journal on Selected Area in Communications*, 1987.

[4] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi radio multi channel wireless mesh networks. In *ACM Mobicom*, 2005.

[5] P. Kyasanur and N. Vaidya. Capacity of multi channel wireless networks: Impact of number of channels and interfaces. In *ACM Mobicom*, 2005.

[6] 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 and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, pages 463–474, Sept., 2005.

[7] W. Wang and X. Liu. A framework for maximum capacity in multi channel multi radio wireless networks. In *Proceedings of CCNC*, 2006.

[8] Mansoor Alicherry, Randeep Bhatia, and Li (Erran) 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.

[9] P. Kyasanur and N. Vaidya. Routing and interface assignment in multi channel multi interface wireless networks. In *WCNC*, 2005.

[10] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In *ACM Mobile Computing and Communications Review(MC2R)*, volume 8, April 2004.

[11] A. Raniwala and T. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *IEEE Infocom*, 2005.

[12] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi radio unification protocol for ieee 802.11 wireless networks. In *Microsoft Research TR*, volume 44, 2003.

[13] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-hop multi-radio wireless mesh networks. In *ACM MobiCom*, Philadelphia, PA, September 2004.

[14] J. Tang, G. Xue, and W. Zhang. Interference aware topology control and qos routing in multichannel wireless mesh net works. In *ACM Mobihoc*, 2005.

[15] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh. Exploiting partially overlapped channels in wireless networks: Turning a peril into an advantage. In *IMC*, 2005.