# Topology Control and Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks

Anjum Naveed, Salil S. Kanhere and Sanjay K. Jha
School of Computer Science and Engineering
The University of New South Wales, Australia
(anaveed,salilk,sjha)@cse.unsw.edu.au

***Abstract*—The aggregate capacity of wireless mesh networks can be improved significantly by equipping each node with multiple interfaces and by using multiple channels in order to reduce the effect of interference. Efficient channel assignment is required to ensure the optimal use of the limited channels in the radio spectrum. In this paper, a Cluster-based Multipath Topology control and Channel assignment scheme (CoMTaC), is proposed, which explicitly creates a separation between the channel assignment and topology control functions, thus minimizing flow disruptions. A cluster-based approach is employed to ensure basic network connectivity. Intrinsic support for broadcasting with minimal overheads is also provided. CoMTaC also takes advantage of the inherent multiple paths that exist in a typical WMN by constructing a spanner of the network graph and using the additional node interfaces. The second phase of CoMTaC proposes a dynamic distributed channel assignment algorithm, which employs a novel interference estimation mechanism based on the average link-layer queue length within the interference domain. Partially overlapping channels are also included in the channel assignment process to enhance the network capacity. Extensive simulation based experiments have been conducted to test various parameters and the effectiveness of the proposed scheme. The experimental results show that the proposed scheme outperforms existing dynamic channel assignment schemes by a minimum of a factor of 2.**

## I. INTRODUCTION

Wireless Mesh Networks (WMN) are emerging as the key future technology for providing wireless broadband access. The capability of self-organization and self-configuration have made WMNs a promising technology for numerous applications like broadband home networking, enterprise networking, neighborhood gaming and more. Several WMN deployments have been planned for major cities across the globe (Taipei, Moscow, Philadelphia, etc). To meet the ever increasing demand for more bandwidth, WMN nodes are equipped with multiple radios (NICs), each operating on different channel. Given the limited number of channels available in the radio spectrum, an effective channel assignment strategy is required. In recent years, a number of channel assignment schemes have been proposed for Multi-Radio Multi-Channel WMN (MR-MC WMN) [1]–[8]. However, the majority of these schemes suffer from one or more of the following drawbacks.

Firstly, the issue of constructing the underlying network topology of the mesh backbone to ensure connectivity – referred to as topology control – is implicitly handled in the process of assigning channels to the radios. This approach has a major drawback that any change in the channel assignment is likely to render certain links to be non-existent. Consequently, flows that are utilizing these links are disrupted and need to be re-routed, which in turn impacts the network throughput. The effect of these disruptions can be significant if these changes are frequent, which is often the case in the majority of the aforementioned schemes, wherein the channel assignments are altered in response to variations in traffic conditions or interference. Secondly, the existing schemes do not take advantage of possible multiple paths that may exist between source and destination nodes. Multipath routing protocols (e.g., AOMDV [9]) can split the traffic over alternate feasible paths thus improving the network capacity.

Another practical issue that is often overlooked is support for broadcasts (and multicasts). Note that, the broadcast primitive is not only important for certain applications (e.g., neighborhood gaming, video-on-demand), but it also plays a vital role in operation of several protocols (specifically MAC and routing protocols). With most channel assignment strategies, there is no single channel that is common to all nodes within an interference domain. Consequently, a node must transmit the same packet on all channels sequentially in order for the broadcast to be received by all the neighboring nodes. This is obviously wasteful due to multiple transmissions involved. On the contrary, schemes like [5], [7] propose the use of a common channel throughout the network to handle broadcast communications. However, this particular channel often gets overloaded in parts of the network where the traffic load is high, consequently affecting the network throughput.

Finally, another pitfall is the inaccuracy and excessive overheads associated with the interference estimation techniques employed in existing dynamic channel assignment schemes. Interference estimation is used to associate a cost with each channel so that the channel assignment algorithm can select the appropriate channel with least cost.

In this paper, we take a holistic approach to address the issues raised above. We present a Cluster based Multipath

1-4244-1455-5/07/$25.00 © 2007 IEEE

Topology control and Channel assignment scheme (CoMTaC) with the primary objective of maximizing network capacity while minimizing the interference and taking advantage of multiple paths in the underlying network topology. We employ the approach that explicitly creates a separation between the topology construction and channel assignment functions, thus minimizing flow disruptions. The first phase of CoMTaC employs a two step topology control scheme that is initiated during network startup. In first step, the constituent nodes are grouped into clusters of small radii (in terms of hop distance). Within each cluster, a common channel (*default channel*) is used by all member nodes on one of their interfaces (*default interface*). Nodes bordering multiple clusters have their second interface tuned to the default channel of highest priority cluster, resulting in inter cluster connectivity. The use of a default channel within the cluster and the inter cluster connectivity provides an efficient broadcasting facility that incurs significantly low overheads. The second step of the topology control scheme aims at identifying multiple feasible paths (in terms of hop distance and interference), thereby enhancing the initial bare-bones connectivity established in the first step. We employ a technique that constructs the *spanner* of the underlying network graph and makes use of the non-default interfaces of each node for establishing the alternate paths. The resulting multiple paths can be exploited to improve the network capacity.

Once the network topology has been constructed as outlined above, the second phase of CoMTaC focuses on channel assignment. A new interference estimation mechanism is proposed which measures the interference with relatively higher accuracy and has lower overheads in comparison to existing mechanisms. Our scheme takes into account, the interference from external sources (i.e. neighboring wireless network deployments) to decide on the allocation of the default channels within the clusters. To measure the interference experienced by the non-default channels, we make use of the average link layer queue length within the interference domain. We also incorporate the concept of partially-overlapping channels ((see [10] for a detailed overview of partially overlapping channels) to further improve spatial reuse and enhance network throughput. We evaluate the proposed scheme through simulations in Qualnet simulator. We report a minimum improvement by a factor of 5 in the network capacity over the base case where all radios are tuned to a single channel. In comparison to existing channel assignment schemes, [2], [5], CoMTaC demonstrates a 200% upturn in the aggregate throughput.

The rest of the paper is organized as follows. Section II reviews related work. Section III introduces a few necessary definitions and formulates the problem. Following this we present the proposed solution, CoMTaC, with Section IV detailing a two-step topology control scheme and Section V elaborating on the channel assignment mechanisms. A thorough simulation-based evaluation of CoMTaC is presented in Section VI. Finally, Section VII concludes the paper.

## II. RELATED WORK

Given the fairly large body of related work in this research domain, we attempt to classify this into three broad categories. We first present current approaches for topology control in WMN. The second part provides an overview of related work in interference estimation. Finally, we present a brief overview of channel assignment schemes.

### A. Topology Control

The problem of topology control has been studied extensively for wireless ad hoc networks [11], [12] where the approaches involve careful tuning of the node transmit power to construct interference optimal topologies. On the contrary, mesh nodes are usually assumed to be transmitting using fixed transmission power. Further, in a typical ad-hoc network, all nodes are equipped with single radio operating on the same channel. On the other hand, in MR-MC WMN, topology control is interlinked with channel assignment in many ways. The problem of topology control in WMN has implicitly been addressed in conjunction with channel assignment [1], [2], [5]. However, the resulting topologies in these schemes do not take advantage of the multiple paths that may exist in the underlying network. Tang et al. [3] have proposed a centralized static channel assignment algorithm, which computes the $K$-connected topology with minimal interference. Marina et al. [4] have proposed a static centralized greedy heuristic channel assignment algorithm for finding the connected low interference topologies in WMN. However, the static schemes do not adapt to the changing traffic conditions. Further, both these schemes incur significant overheads to realize broadcasts, due to the unavailability of a common channel in the entire network.

### B. Interference Estimation

The concept of estimating interference and adapting to the changing traffic conditions through channel assignment has been explored in [1], [2], [6], [8], wherein the traffic load is used as a metric to identify interference. However, these schemes require the a priori knowledge of the traffic demand of each node, which may not always be possible. The number of interfering radios has been used as a measure of interference in [3], [4]. However, this metric is not always accurate, since the traffic load on each radio is neglected. De Couto et al. [13] have proposed the ETX (Expected Transmission Count) metric, which measures the probability of successful transmission of data packets over a link, for estimating the link quality. However, ETX cannot provide a measure of the traffic demand (i.e. load) on a particular channel within the interference domain, which is necessary for efficient channel assignment. Ramachandran et al. [5] propose that WMN nodes periodically estimate the channel interference by switching one interface to the packet capturing mode and sensing the load on the channel. Expected Transmission Time (ETT) [14] (based on ETX) is used as the supporting metric. The process is repeated sequentially for each channel. Though this approach exhibits higher accuracy in estimating interference, it renders one radio interface unavailable for a relatively long duration of time.

### C. Channel Assignment

A number of channel assignment schemes have been proposed in recent years [1]–[8]. Use of multiple radios and multiple channels in WMN was first proposed by Raniwala et al. [1] and a centralized channel assignment scheme was outlined. In a subsequent publication [2], the authors proposed a dynamic distributed channel assignment and routing algorithm. However, both these schemes rely on prior availability of the traffic demands of each mesh node, which is not always feasible. Alicherry et al. [6] proposed a centralized load-aware link scheduling, channel assignment and routing protocol. The authors propose the division of fixed duration time frames into slots where a specific set of nodes can transmit within each time slot on specific channels assigned by a channel assignment algorithm. The centralized nature of the proposed algorithm and the assumption of infrequent changes in traffic demands makes the proposed solution less attractive. In addition, all of the aforementioned schemes incur significant overheads to effectuate broadcast transmissions, due to the lack of a common channel.

Ramachandran et al. [5] have proposed a centralized channel assignment algorithm where a default interface for each node is tuned to a common default channel, which significantly simplifies broadcast transmission. However, given that the default channel is utilized in the entire network, it is quite likely that this channel can experience significant interference, particularly in parts of the network, where the traffic load is very high. Localized use of default channel within an interference domain can be an attractive alternate which we explore in this paper. Mohsenian et al. [8] have proposed the use of partially overlapping channels in their load-aware channel assignment algorithm. The authors have used the TCP congestion control mechanism to estimate the load (and interference). However, this does not capture UDP traffic, which is a significant component of today's Internet traffic. Mishra et al. [10] have formally modeled the degree of overlap between partially overlapping channels. The authors have modified the channel assignment algorithms proposed in [6] to incorporate the partially overlapping channels and shown the improvements as compared to the original algorithms. We have used this model in our work to take advantage of partially overlapping channels.

## III. PROBLEM FORMULATION

It is a common practice to model a network as a connectivity graph. We follow the same approach and model the WMN as a graph in multi-dimensional space (e.g., 3D). We first introduce some definitions that are important for our scheme and then formally define the problem that is addressed in the paper.

### A. Network Model

Let the undirected nonplaner multigraph $G(V, E)$ represent the WMN where $V \rightarrow \mathbf{R^d}$ (s.t. $|V| = n$ & $d \geq 1$) is the set of vertices and each $v \in V$ corresponds to a particular node in WMN. $E$ is the set of edges representing the wireless links between WMN nodes. In the rest of the paper, we use the terms (i) *vertex* and *node* and (ii) *edge* and *link* interchangeably. Each node is equipped with $k$ radio interfaces ($k \geq 2$) one of which is designated as the *default* interface. Note that, each node may have different number of interfaces. Let $G_T(V, E')$ represent the graph induced by the topology control scheme and $G_A(V, E")$ represent the graph induced by the channel assignment scheme where $G_A \subseteq G_T \subseteq G$.

### B. Transmission and Interference Model

Let $R_t$ and $R_i$ denote the fixed transmission range and interference range of all the wireless interfaces respectively, where $R_i > R_t$. Let $dist(u, v)$ represent the Euclidean distance between two nodes $u, v \in V$. For two nodes $u, v \in V$, direct communication is only possible if the $dist(u, v) \leq R_t$ and at least one of the interfaces of the two nodes operates on a common channel. We assume that the wireless links are symmetric, such that for $u, v \in V$, if $u$ can transmit to node $v$ then $v$ can also transmit to node $u$. Two links $e1 = (u1, v1)$ and $e2 = (u2, v2)$ interfere with each other if both edges operate on a common channel and any of the distances $dist(u1, u2), dist(u1, v2), dist(v1, u2), dist(v1, v2) \leq R_i$.

*Definition 1:* The interference $I(e)$ experienced by the link $e \in E$ is the sum of the traffic load on all the interfering links.

$$I(e) = \sum Load(i)$$

*where $i$ is an interfering link*

In the proposed scheme (CoMTaC), the topology construction is performed during the network initialization phase when no user traffic is present in the network. Therefore, we assume unit load for all the links in the WMN and the simplified definition of link interference derived from Definition 1 is used.

*Definition 2:* The interference $I_T(e)$ experienced by the link $e \in E$ is the number of interfering links.

$$I_T(e) = \textit{Number of Interfering Links}$$

We can extend the definition of link interference to the definition of path interference (which is used as cost metric for the sake of topology control) as follows.

*Definition 3:* The interference experienced by the unit flow on a path between the nodes $u, v \in V$ is the sum of link interference values of all the links on that path.

$$Cost(u, v) = I_T(Path(u, v)) = \Sigma\, I_T(e)$$
$$\forall\, e \textit{ on path (u,v)}$$

### C. Problem Definition

As discussed earlier, our proposed scheme seeks to create a clear demarcation between the topology control and channel assignment functionalities. Thus, our problem formulation is made up of two parts:

*Part 1: Given a network connectivity graph $G(V, E)$, construct the network topology graph $G_T(V, E')$ such that the network* *connectivity is ensured, overheads for broadcasting are minimal and multiple paths (of feasible cost) exist between source-destination pairs using the edges, which result in minimum interference.*

*Part 2: Given the above network topology graph $G_T(V, E')$, assign the channels from the set of available channels, to the links $e \in E'$ such that the interference $I(e)\ \forall\ e \in E'$ is minimized.*

## IV. NETWORK TOPOLOGY CONTROL

Topology control in CoMTaC is achieved via a two-step process, with the final outcome being the connected multigraph $G_T(V, E') \subseteq G(V, E)$ representing the network topology. In first step, the nodes are grouped into small clusters with the objective of designating a common *default channel* within each cluster, which is assigned to the *default interface* of each constituent cluster node. A simple mechanism to interconnect clusters using the border nodes is proposed. In second step, a spanner construction algorithm is used to identify multiple feasible paths using *non-default interfaces*, enhancing the basic connectivity established in first step.

### A. STEP 1: Cluster Construction

Algorithm 1 provides the pseudo-code for cluster construction which is based on the clustering technique of Gonzalez [15]. The algorithm has low time complexity of $O(kn)$ and uniform sized clusters are constructed. The cluster radius, $r$, (maximum hop distance from the cluster head) is a design parameter. The value of $r$ should be chosen keeping in mind the trade-off between the size of the cluster (i.e., number of nodes) and the number of clusters formed. The value of $r = 2$ is recommended because in most realistic mesh networks, the interference domain of a node typically covers 2-hop neighborhood [2]. The input to the algorithm is the graph $G(V, E)$ (where each $v \in V$ has knowledge of its hop distance from gateway) and the set of gateway nodes (nodes connected to the Internet). Initially each gateway node is designated as a cluster head and all the nodes connected to this gateway become part of one cluster (lines 8-13 of procedure *Cluster*). Two such clusters are shown in Figure 1 (clusters encircled by solid line). However, due to limited number of gateway nodes, the radius of these clusters can be large. Therefore, the procedure *ConstructCluster* is repeatedly invoked to construct a new cluster from the existing clusters (Lines 14-16) until the radius of each cluster is reduced to $r$ at the most. To construct a new cluster, the node $v \in V$ with maximum hop distance from cluster head is selected as candidate for new cluster head (Line 19). The cluster is constructed around the newly selected cluster head by adding the nodes $u \in V$ into the cluster, for which the hop distance to new cluster head is lesser than the distance from the current cluster head (Lines 20-30). These nodes are now removed from the clusters, which they were previously part of. In Figure 1, the nodes encircled by the dotted line become part of the new cluster with the cluster head denoted by 'CHnew'. We now present a discussion about how network connectivity is maintained and broadcast is handled.

**Algorithm 1** The Clustering Algorithm

```
INPUT: Graph $G(V,E)$,   $|V| = n$ & Set $V_G$ of gateways & $\forall v \in V$, $hopcount(v)$ is hop distance of $v$ from closest gateway with gateway ID $GID(v)$ & $r$
OUTPUT: Set $\chi$ containing cluster sets $X_i s.t. \forall v \in V, v$ is element of some $X_i$.
 1: procedure CLUSTER
 2:     $m \Leftarrow |V_G|$    $\chi \Leftarrow \emptyset$
 3:     $p \Leftarrow m$                                  ▷ $p$ is number of clusters
 4:     for $i \Leftarrow 1$ to $m$ do
 5:         $X_i \Leftarrow v_i$    $v_i \in V_G$
 6:         $\chi \Leftarrow X_i$
 7:     end for
 8:     for $i \Leftarrow 1$ to $n$ do
 9:         $x \Leftarrow GID(v_i)$    $v_i \in V$
10:         $X_x \Leftarrow X_x \cup \{v_i\}$
11:         $clusterdist(v_i) \Leftarrow hopcount(v_i)$
12:         $CHID(v_i) \Leftarrow GID(v_i)$
13:     end for
14:     while $clusterdist(x) > r$ for any $x \in \{X_1 \cup X_2 ... \cup X_p\}$ do
15:         ConstructCluster($\chi, p, n, V$)
16:     end while
17: end procedure

18: procedure CONSTRUCTCLUSTER($\chi, p, n, V$)
19:     $h \Leftarrow x$ s.t. $clusterdist(x)$ is maximum $\forall x \in \{X_1 \cup X_2 ... \cup X_p\}$
20:     $p \Leftarrow p + 1$
21:     $X_p \Leftarrow \{h\}$
22:     $CHID(h) \Leftarrow h$
23:     $X_{CHID(h)} \Leftarrow X_{CHID(h)} \setminus \{h\}$
24:     for $i \Leftarrow 1$ to $n$ do
25:         if $clusterdist(v_i) > dist(h, v_i)$ then
26:             $X_p \Leftarrow X_p \cup \{v_i\}$
27:             $X_{CHID(v_i)} \Leftarrow X_{CHID(v_i)} \setminus \{v_i\}$
28:             $CHID(v_i) \Leftarrow h$
29:         end if
30:     end for
31: end procedure
```

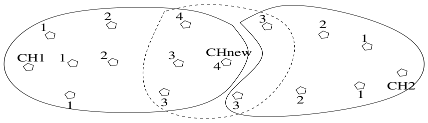


Fig. 1. Cluster construction. Number against each node is hop distance from cluster head. Dotted line encircles newly constructed cluster.

*1) Network Connectivity:* Within each cluster, the default interface of each node is configured on a common default channel, which is assigned using the channel assignment scheme discussed in Section V. This ensures the connectivity within the cluster. Inter cluster connectivity is established by assigning the default channel of the cluster with least cluster head ID to one of the non-default interfaces of the border nodes in the neighboring cluster. This is shown in Figure 2 where the border nodes $I$ and $L$ will configure one of the non-default interfaces to channel 1 (default channel of cluster 1 with least CHID). Node $N$ will configure one of its non-default interfaces to channel 2. Note that node $N$ does not have any neighbors from cluster 1, therefore, it does not configure its interface to channel 1. It is now easy to prove that Algorithm 1 ensures connectivity by utilizing the default interfaces only (except edge nodes where one non-default interface might be consumed).

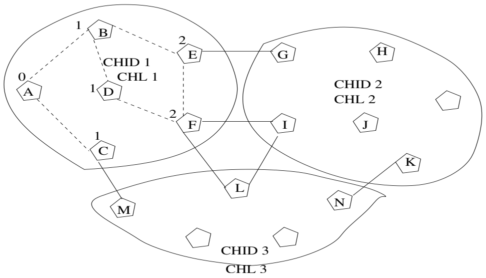


Fig. 2. Inter-cluster connectivity using border nodes

*2) Broadcast Transmissions:* Since the nodes within the same cluster have a common default channel, broadcast transmission within the clusters is possible without extra overhead. However, to ensure that the broadcast can take place beyond the boundary of the cluster, certain border nodes need to broadcast on two interfaces: the default interface and the interface, which is configured to the default channel of the neighboring cluster (e.g., node $N$ in Figure 2). This technique is efficient because of limited multiple transmissions as compared to schemes such as [1], [2], [6] where each node is required to broadcast on all channels being used in neighborhood.

### *B. STEP 2: Spanner Construction*

The second step of the topology control scheme of CoMTaC constructs the spanner of the network connectivity graph to identify multiple feasible paths in the WMN by utilizing the non-default interfaces. Link and path interference (see Definition 2 & 3) are used as cost metrics. A path between nodes $u, v \in V$ is feasible if the path cost $cost(u, v)$ is a constant factor of the lowest possible path cost $cost_{SP}(u, v)$ between these nodes. To be more precise, the spanner of a graph $G$ is formally defined as,

*Definition 4:* A connected graph $G_S(V, E')$ is a spanner of the graph $G(V, E)$ if $E' \subseteq E$ and for any two vertices $u, v \in V, \quad cost_{G_S}(u, v) = t * cost_G(u, v)$ for $t \geq 1$.

The value of $t$ in Definition 4 is a design parameter. A large value of $t$ results in more alternate paths, however, the cost of such paths can be high. In addition, increasing the value of $t$ beyond a certain upper bound does not create more loop-free paths. Therefore, smaller values (e.g., 2, 3) are recommended, which results in multiple paths using lesser number of links with lower values of interference. Note that, the spanner is also constructed for topology control in ad hoc networks, with the objective of minimizing interference (e.g., [16]). However, in most cases, a variation of the basic Minimum Spanning Tree (MST), instead of the spanner graph, is constructed. The main drawback of the MST is that, it does not take advantage of the inherent multiple paths that exist in WMN.

**Algorithm 2** Spanner Algorithm

```
INPUT: $G(V, E)$  $|V| = n$ & $\forall v \in V, N(v)$ is set of $t/2$-hop neighbors
    of $v$ & Cost(u,v) defined in Definition 3 & $t$.
OUTPUT: Spanner graph $G_S(V, E')$ of $G$
 1: $E' \Leftarrow \emptyset$      $G_S \Leftarrow (V, E')$
 2: for $i \Leftarrow 1$ to $n$ do
 3:     $u \Leftarrow v_i$      $v_i$ is $ith$ element of $V$
 4:     for $j \Leftarrow 1$ to $|N(v_i)|$ do
 5:         $v \Leftarrow p_j$      $p_j \in N(v_i)$
 6:         if for some $w \in V, Cost(u, w) + Cost(w, v) < t * Cost(u, v)$
    then
 7:             $E' \Leftarrow E' \cup \{edges\ on\ path(u, w)\} \cup \{edges\ on\ path(w, v)\}$
 8:         else
 9:             $E' \Leftarrow E' \cup \{(u, v)\}$
10:         end if
11:     end for
12: end for
```

Algorithm 2 constructs the spanner of the network graph $G$ in the following manner. For every edge $(u, v) \in E$, if a path with lesser path interference exists between $u$ and $v$ as compared to $t$ times the interference on this edge, then the path is added into the resulting topology, otherwise the edge itself is added. Intuitively, the edges with high value of interference are removed from the network. Note that, this algorithm is distributed in nature and lines 4-11 can be executed at each node independently.

The spanner construction is followed by neighbor-interface binding. Such a binding is necessary for channel allocation because at least one of the non-default interfaces of the two neighboring nodes (in the spanner graph) should be on the same channel in order for the link between the nodes to exist. A simple strategy of distributing the neighbors equally among non-default interfaces can lead to a channel dependency problem, as illustrated by the example involving nodes $A, B, E, G$ in Figure 2. Suppose node $E$ uses interface 2 for links $BE$ and $EG$ and node $B$ uses interface 1 for links $AB$ and $BE$. Suppose the channel assignment of link $AB$ is changed. This essentially means that interface 1 of node $B$ will be switched to new channel, forcing node $E$ to change its interface 2 to the same channel in order for link $BE$ to exist. Similarly, node $G$ will need to change its channel assignment in order for link $EG$ to exist. Therefore, a change in channel assignment for one link can trigger a series of changes in the neighborhood. CoMTaC limits such a dependency to one hop neighbors using the following constraints:

- The non-default interface used by the border node of a cluster to bind the neighbors in the neighboring cluster should not be used to bind the neighbors in its own cluster.
- The non-default interface used by a node for neighbors of hop distance (from gateway) less than or equal to its own hop distance should not be used to bind the neighbors with hop distance more than the node itself.

Consider the same set of nodes in Figure 2, where the number against each node shows its hop distance from the gateway. Suppose node $E$ is using interface 1 for link $EG$. This interface cannot be used for link $BE$ or $BF$ based on the first constraint, because $G$ is a node in the neighboring cluster. Further suppose that node $B$ uses interface 1 for link $AB$. This interface cannot be used for link $BE$ based on the second constraint, because node $A$ has lower hop distance, while node $E$ has greater hop distance than $B$. However, Node $B$ can use same interface for link $BD$. Now consider a change in the channel assignment for link $AB$. This change will force node $B$ to switch interface $1$ to a new channel and inturn triggers a change in node $D$. However, this change is only limited to the connected nodes with hop distance $0$ and $1$. Note that, for certain nodes with lesser number of interfaces (e.g., only one non-default interface) it may not be feasible to fulfill these constraints. The links incident on such nodes are removed from the edge set $E'$ (graph $G_T(V, E')$) to achieve the edge set $E''$ (graph $G_A(V, E")$ defined in Section III-A) which is the feasible edge set (graph) for channel assignment.

## V. CHANNEL ASSIGNMENT

The second phase of CoMTaC focuses on assigning the channels to the links of the graph $G_A(V, E'')$, which is constructed during the first phase. Channel assignment is preceded by the process of interference estimation, which aims at associating a cost with each channel, thus enabling selection of the best quality channel. The estimation process is executed locally at each node. For the selection of default channel within each cluster, the interference from external sources (e.g., external WiFi deployments) is taken into account. The only way to accurately measure external interference is by periodic passive monitoring of traffic load on the channels, as proposed in [5]. We employ a variant of this technique in our scheme, which distributes the monitoring task among the nodes within a cluster. In assigning channels to the non-default interfaces, we propose the use of the average link layer as a metric for estimating the interference on the candidate channels. Note that, in our scheme we assume that the cluster head has complete information about the nodes and their neighbors within the cluster.

### *A. Default Interface Interference Estimation and Channel Assignment*

The interference estimation process for the default channel of a cluster is collaboratively managed by the constituent nodes. Here we focus primarily on the interference created by external sources. Our scheme is based on the passive monitoring mechanism proposed in [5]. Two parameters, channel utilization and channel quality are used as metrics. One of the non-default interfaces of the nodes is configured periodically (every $T_E$ units of time, with $T_E$ being reasonably large to avoid frequent changes) to the packet capture mode for a specific interval of time ($T_C$ units of time) on each channel. Note that the packets captured include the packets from external networks as well as those from nodes in WMN. The packets captured during this interval are used to calculate channel utilization. However, the captured traffic on a particular channel may not be representative of the actual load on this channel, for example, the number of successful transmissions will be low if the interference is high. Therefore, channel quality is used to refine the estimation results. The channel quality is based on a number of lower-layer metrics such as bit or frame error rate, received signal strength, etc. Almost all commodity wireless cards measure the channel quality and this value is accessible through software packages like *iwconfig* [17]. However, this estimation process is expensive due to the channel switching delay involved and also affects the network capacity since the interface is unavailable for forwarding the traffic during estimation. Therefore, for each period $T_E$, the cluster head selects only a few nodes within the cluster using the following steps:

- `STEP 1`: Select a node randomly, add to set $S$.
- `STEP 2`: Mark the node and its one hop neighbors.
- `STEP 3`: Repeat STEP 1 & 2 over unmarked nodes in the cluster until all nodes are marked.

The nodes that belong to the set $S \quad (|S| = s)$ perform estimation for the current period. The random selection of nodes results in distribution of the overhead. Further, to avoid disruption of traffic flows, all flows that use the non-default interface, which is chosen for monitoring, are redirected to the default interface. Each node that performs the estimation, transmits the value of both metrics for each channel to the cluster head. The cluster head computes the best channel as follows. Let $C \quad (|C| = m)$ be the set of available channels. Let $U_{ij}$ and $CQ_{ij}$ respectively represent the utilization and channel quality measured by the node $i$ on channel $j$. The cluster head combines the collected information to calculate the cost of a particular channel using the following equation, where $\alpha$ is a weight factor,

$$Cost_j(default) = (1-\alpha)\sum_{i=1}^{s} U_{ij} + \alpha\frac{1}{\sum_{i=1}^{s} CQ_{ij}} \quad \forall j \in C$$

The channel with the least cost is selected by the cluster head as the default channel. If this channel if different from the currently used default channel, the cluster head informs all the nodes within the cluster of the new default channel and nodes configure their default interfaces to this channel.

### *B. Non-default Interfaces Interference Estimation and Channel Assignment*

Relatively more accurate and inexpensive estimation is possible for the interference from within the network as compared to the interference from external sources. Research shows that channel utilization in conjunction with channel quality serve as sufficient metrics for interference estimation [5], [13], [14]. However, techniques that are currently employed to estimate the value of these metrics are expensive (see Section II-B). In CoMTaC, we propose to use the average link layer queue length as a metric for interference estimation which is easily accessible from the link layer. It is easy to show that a large value of the average queue length of the interfaces operating on a particular channel is always indicative of high interference, irrespective of the cause of increase in queue length (e.g., congestion in upstream network).

Each node periodically (every $T_A$ units of time) transmits the information about channel and the queue length for all interfaces (default as well as non-default) to the cluster head. The border nodes used for inter-cluster connectivity transmit the information to the cluster head of neighboring connected cluster, in addition to their own cluster head. The information from neighboring cluster nodes and default interfaces is only used for computation and no channel is assigned to these interfaces. The cluster head performs the following computation for non-default interfaces of each node. Let $C \quad (|C| = m)$ be the set of available channels. For a particular node $v$, Let $A_v = a_{ij}$ ($a_{ij} = 1$ if interface $j$ operate on channel $i$ and 0 otherwise) be the matrix of order $m \times n$, where $m$ is the number of channels and $n$ is the number of interfering interfaces with node $v$ including its own interfaces. Let $Q_{max}$ be the maximum possible queue length and $Q = q_i$ be the column matrix with dimension $n$ such that $q_i = \frac{\text{Avg. Queue Len. of interface } i}{Q_{max}}$. Let $Cost(non - dflt) = c_i$ be the column matrix with dimension $m = |C|$ where $c_i$ is the cost of using channel $i$. Using the average queue length as metric, the cost matrix can be given as:

$$Cost(non - dflt) = A * Q \quad \text{OR} \quad c_i = \sum_{j=1}^{n} a_{ij} * q_j \tag{1}$$

The above equation calculates the cost of channels if the channels are non-overlapping. However, partially overlapping channels interfere with one another depending upon the degree of overlap. We use the I-factor defined in [10] to capture the effect of overlap (discussion on the I-factor is omitted for brevity). For any two channels $i, j$ the value of I-factor is a constant such that $0 \leq I(i, j) \leq 1$. $I(i, j) = 1$ if $i = j$ and $I(i, j) = 0$ for non-overlapping channels. Let $X = x_{ij}$ be the matrix of order $m \times m$ such that $x_{ij} = I(i, j)$, then the Cost matrix defined in Equation 1 can be updated as follows.

$$Cost(non - dflt) = X * A * Q \quad \text{OR} \quad c_i = \sum_{k=1}^{m} x_{ik} * (\sum_{j=1}^{n} a_{ij} * q_j) \tag{2}$$

Based on the channel costs, the cluster head computes the new channel assignment for the interfaces of all the cluster nodes. Algorithm 3 is used to assign the channels to the non-default interfaces of the nodes within the cluster. The node-interface pairs of the form $(v, i)$ are considered for assignment. First priority is given to the interfaces of the border nodes used for inter-cluster connectivity (Lines 1 -17 of the algorithm 3). If the interface $i$ of the border node $v$ is connected to a higher priority cluster, then this interface is assigned the channel used by its neighboring nodes from that cluster (nodes $G, I, L, M, N$ in Figure 2). Otherwise, a new channel assignment is calculated for this interface of the node. If the best channel calculated differs from currently used channel, the node (nodes $E, F, K$) and neighbors of node bound to this interface (for node $F$, neighbors $I, L$) are informed to update the assignment. The node-interface pair is then removed from set $U$.

**Algorithm 3** Non-default Channel Assignment Algorithm

```
INPUT: Set $U = \{(v, i, c, q) | v \in V$, channel $c$ is used on interface $i$ of $v$ with queue length $q\}$ sorted by hop distance and number of neighbors per interface.
OUTPUT: Channel assignment of the nodes updated.
1:  $E \Leftarrow \{(v, i) | (v, i, c, q) \in U \& v$is an edge node$\}$
2:  while $E \neq \emptyset$ do
3:      $(v, i) \Leftarrow$ Element of $E$ with min. $hopcount(v)$.
4:      if $i$ is bound to nodes of neighboring cluster then
5:          if CHID of neighboring cluster is greater. then
6:              Assign $i$, the channel assignment from its neighbor assignment.
7:          else
8:              Calculate new assignment for (v,i)
9:              if Channel assignment needs to be updated. then
10:                 Inform $v$ of new assignment for $i$.
11:                 $v$ updates assignment of $i$ and informs bound neighbors of the assignment.
12:             end if
13:         end if
14:         $U \Leftarrow U \setminus \{(v, i, c, q)\}$
15:     end if
16:     $E \Leftarrow E \setminus \{(v, i)\}$
17: end while
18: while $U \neq \emptyset$ do
19:     $(u, j) \Leftarrow$ First element of $U$
20:     Calculate new assignment for (u,j)
21:     if Channel assignment needs to be updated. then
22:         Inform $u$ of new assignment for $j$.
23:         $u$ updates assignment of $j$ and informs bound neighbors of the assignment.
24:         Bound neighbors update assignment for respective interfaces
25:     end if
26:     $U \Leftarrow U \setminus \{\{(u, j, c, q)\} \cup (v, i, c, q) | (v, i)$is bound to$(u, j)\}$
27: end while
```

The remaining interface-node pairs in set $U$ are processed as follows. The highest priority interface (with most number of neighbors) of the highest priority node (least hop distance from gateway) is selected and channel assignment for the interface is calculated (Node $A$ in Figure 2 selected first). If the best channel calculated differs from the currently used channel, the node and the bounded neighbors with the interface are informed to update the assignment (Lines 19 - 25). The node-interface pair along with the bounded node-interface pairs are removed from the set $U$ (Line 26), ensuring that the interfaces of the nodes that are connected with higher priority neighbors are not re-processed.

## VI. SIMULATION-BASED EVALUATION

The effectiveness of CoMTaC scheme is evaluated through simulation based experiments using the Qualnet discrete event simulator. We first evaluate the topology control scheme of CoMTaC. We investigate the effect of the design parameters, $r$ and $t$ on the cluster size, number of clusters and average hop distance. Following this, we evaluate the effectiveness of channel assignment scheme employed by CoMTaC and compare its performance with other popular algorithms such as Hyacinth [2] and [5] (the authors refer to this as BFS-CA). We also include the case where all radios are tuned to a single channel as the base case, which we refer to as *Single Channel*.

### A. Topology Control Evaluations

Recall from Section IV-A that $r$ is the design parameter that governs the size of the cluster. Different values of $r = 2, 3, 4$ with sparse (average internode distance of 125 meters) and dense (Average distance of 50m) node placement are considered to evaluate the impact on cluster size. One gateway nodes is used in all cases. Figure 3(a) shows minimum, maximum and average cluster size for $r = 2$ and $4$ with different number of nodes in the network. The difference between minimum and maximum cluster size for most of the cases is at the most 4 showing that uniform sized clusters are constructed. Further, for $t = 2$ the average cluster size of 8 for sparse deployment and 15 for dense deployment shows that small clusters are constructed. However, for $t = 4$, the average cluster size increases considerably, and the benefit of clustering (to localize the use of default channel) is almost nullified.

Spanner construction is evaluated for different values of parameter $t$. Figure 3(b) shows the value of average hop distance for different number of nodes and values of $t$. All pair shortest path average hop distance is also shown for comparison. Results show that the average hop distance increases with the increase in value of $t$. This is because certain links with higher interference values are deleted using comparison on line 6 of Algorithm 2. It was also observed that the average hop distance did not increase with increase in value of $t$ beyond 4. Therefore $t = 4$ served as threshold value in case of $n = 25$ and 36.

### *B. Channel Assignment Evaluations and Comparisons*

In our evaluations, we compare the performance of CoMTaC with that of Hyacinth [2], BFS-CA [5] and *Single Channel*. The two metrics used in our evaluations include: (i) the average number of interfering radios on each channel, which captures the level of interference in the network, and (ii) aggregate throughput. Note that, we include two versions of CoMTaC in the evaluations; *CoMTaC (non-overlapping)* only utilizes non-overlapping channels; whereas *CoMTaC (partially overlapping)* incorporates partially overlapping channels as well. The results of these comparisons are summarized in Figures 3(c)-3(f). IEEE 802.11b is used as the MAC layer protocol, unless otherwise stated. The limited number of channels available in IEEE 802.11b allows to better demonstrate the effectiveness of CoMTaC. Table I lists the rest of the parameters used in the evaluations. A simple multipath routing protocol was also implemented, which was used with CoMTaC and the *Single Channel* case. Hyacinth incorporates a routing protocol in addition to resolving channel assignment, while a modified version of OLSR is used with BFS-CA (the implementation of BFS-CA was obtained from the authors).

Figure 3(c) shows the average number of interfering radios per channel for the different schemes. As seen from Figure 3(c), the *single channel* has all radios tuned to the same channel resulting in a high value for the interfering radios. Hyacinth achieves a marginally lower value (20) as compared to the non-overlapping channel version on CoMTaC (21). This is because CoMTaC uses a common channel within the entire cluster resulting in more number of interfering radios operating on this channel. On the other hand, BFS-CA has a higher value of 23 because a common channel is used throughout the network thereby increasing the number of interfering radios on this channel. Further note that the value for CoMTaC reduces to 13 if partially overlapping channels are included, since this enables greater spatial reuse. The ratio of the values for CoMTaC (non-overlapping) and CoMTaC (partially overlapping) is $\frac{21}{13} = 1.6$, thus demonstrating the increase in spatial reuse when partially overlapping channels are used.

TABLE I
PARAMETERS FOR CHANNEL ASSIGNMENT EXPERIMENTS

| | |
|---|---|
| No. of nodes = 50 | No. of gateways = 2 |
| No of Interfaces per node = 3 | Network Topology = Uniform |
| $r = 2$ | $t = 2$ |
| No. of traffic flows = 5 | Traffic Type = CBR-UDP |
| Data rate for each flow = variable | Simulation time = 15 Minutes |

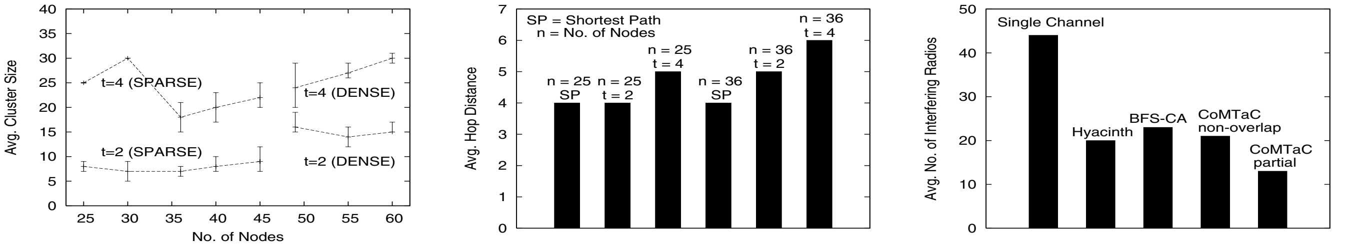
We next evaluate the throughout achieved by the aforementioned schemes as a function of increasing aggregate traffic load. Each experiment was repeated 5 times in the steady state with different source destination pairs for the 5 flows and the results were averaged. We first investigate the performance in the case when there is no external interference.

Figure 3(d) illustrates the aggregate throughput for the different schemes. CoMTaC (non-overlapping) and CoMTaC (partially overlapping) outperform the base case by a factor of 5 and 7 respectively. CoMTaC (non-overlapping) outperforms BFS-CA and Hyacinth by 30% and 80% respectively on average. The improved network capacity with CoMTaC in comparison to BFS-CA is attributed to the lower overheads associated with the interference estimation technique.

Further, the localized use of common channel within each cluster also contributed to better utilization of default interfaces of the nodes. CoMTaC also scores over Hyacinth due to the multipath topology created by the spanner in the topology control phase as opposed to the tree topology that is generated by Hyacinth. The greater spatial reuse offered by the use of partially overlapping channels enables CoMTaC (partially overlapping) to outperform BFS-CA and Hyacinth by a factor of 1.9 and 2.7 respectively.

Recall that 802.11b radios were used in the above comparison. We next investigate if the larger number of channels offered by 802.11a have an effect on the throughput. The above simulations were repeated by replacing the 11b with 11a radios and Figure 3(e) illustrates the ensuing results. The general trend is similar to that with 11b radios, with CoMTaC showing even better performance due to the greater number of channels that are available for use within each cluster.

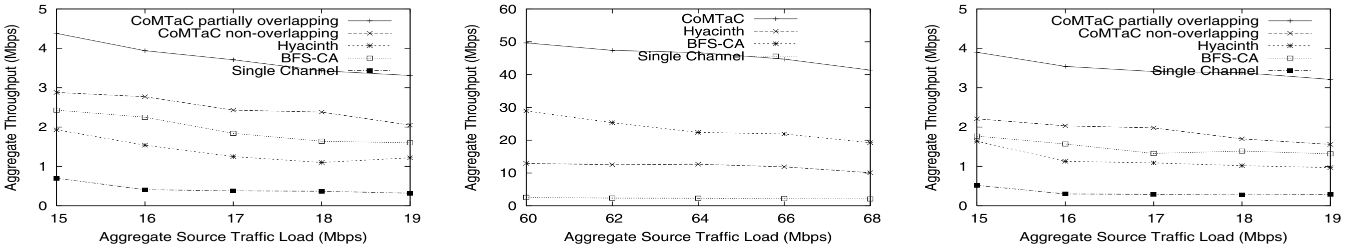
In the second set of experiments, we wish to investigate the effectiveness of CoMTaC in the face of external interference i.e. interference from traffic, which is external to the WMN. To model this we created two flows in the network external to the WMN. Fixed channels (channels 6 and 11 which are typically used in WiFi deployments) were used for these flows. CBR-UDP traffic was used with the aggregate load of these flows equal to about one fifth of the aggregate traffic load within the WMN. Figure 3(f) shows the aggregate throughput comparison. In this case we only illustrate the results with 802.11b radios since we observed similar results for 802.11a as well. One can readily observe that Hyacinth, which does not incorporate any measures to deal with external interference, suffers significantly (as compared to Figure 3(d)). In addition,



(a) Cluster size for different number of nodes with $r = 2$

(b) Avg. Hop distance for different No. of nodes and values of $t$

(c) Average number of interfering radios for different schemes



(d) Throughput comparison with no external interference (IEEE 802.11b radios)

(e) Throughput comparison with no external interference (IEEE 802.11a radios)

(f) Throughput comparison with external interference (IEEE 802.11b radios)

Fig. 3. CoMTaC Analysis

the aggregate throughput with CoMTaC (partially overlapping) only drops fractionally in comparison to that of BFS-CA and CoMTaC (non-overlapping). Note that, CoMTaC (partially overlapping) now outperforms these schemes by a factor of 2.4 and 1.8 respectively (as compared to 1.9 and 1.5 in the case when there is no external interference). This is because both the external flows are using non-overlapping channels (6 and 11), thus limiting the capacity of these channels, which in turns affects all schemes that only use non-overlapping channels.

## VII. CONCLUSION

We proposed a cluster-based multipath topology control and channel assignment scheme (CoMTaC) with independent channel assignment and topology control functions, thus minimizing flow disruptions. The cluster-based topology of CoMTaC ensured basic network connectivity with intrinsic support for broadcast. Multipath topology was constructed which took advantage of the inherent multiple paths that exist in a typical WMN by constructing a spanner of the network graph. The dynamic distributed channel assignment scheme of CoMTaC employed a novel interference estimation mechanism based on the average link-layer queue length within the interference domain. The simulation based experiments showed that CoMTaC outperformed the base case of single channel WMN by a factor of at least 5. The comparison with existing channel assignment schemes showed that CoMTaC outperformed these schemes by a factor of at least 2.

## REFERENCES

[1] A. Raniwala; K. Gopalan; T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mobile Computer Communications*, 8(2):50–65, April 2004.

[2] A. Raniwala; Tzi cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *Proceedings of InfoCom'05*, volume 3, pages 2223–2234, March 2005.

[3] J. Tang; G. Xue; W. Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *In Proceedings of MobiHoc '05*, May 2005.

[4] M.K. Marina; S.R. Das. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. In *2nd International Conference on Broadband Networks*, volume 1, pages 381–390, October 2005.

[5] Krishna N. Ramachandran; Elizabeth M. Belding; Kevin C. Almeroth; Milind M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *InfoCom'06*, April 2006.

[6] M. Alicherry; R. Bhatia; L. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proceedings of MobiCom '05*, September 2005.

[7] P. Kyasanur; N.H. Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *SIGMOBILE Mobile Computer Communications*, pages 31–43, January 2006.

[8] A. H. Mohsenian; V.W.S. Wong. Joint optimal channel assignment and congestion control for multi-channel wireless mesh networks. In *IEEE ICC*, volume 5, pages 1984–1989, June 2006.

[9] S.R. Das A. Nasipuri. On-demand multipath routing for mobile ad hoc networks. In *Proceedings of the ICCCN'99*, October 1999.

[10] Arunesh Mishra; Vivek Shrivastava; Suman Banerjee; William Arbaugh. Partially overlapped channels not considered harmful. In *Proceedings of SIGMETRICS '06/Performance '06*, pages 63–74, June 2006.

[11] Martin Burkhart; Pascal von Rickenbach; Roger Wattenhofer; Aaron Zollinger. Does topology control reduce interference? In *Proceedings of MobiHoc '04*, pages 9–19, May 2004.

[12] Tomas Johansson; Lenka Carr-Motyckova. Reducing interference in ad hoc networks through topology control. In *DIALM-POMC '05*, pages 17–23, September 2005.

[13] D. De Couto; D. Aguayo; J. Bicket; R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MOBICOM '03*, 2003.

[14] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of MobiCom '04*, pages 114–128, September 2004.

[15] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[16] Yu Wang; Xiang-Yang Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *Proceedings of DIALM-POMC '03*, 2003.

[17] http://www.linuxcommand.org/man_ pages/iwconfig8.html. Online linux commands manual.