Spectrum Aware Routing in Cognitive Radio Mesh Networks

Ioannis Pefkianakis[1], Starsky H.Y. Wong[2] and Songwu Lu[1]
Dept. of Computer Science UCLA 4732 Boelter Hall, Los Angeles CA 90025[1]
IBM T.J. Watson Research Center 19 Skyline Hawthorne, NY 10532[2]

{pefkian, slu}@cs.ucla.edu[1] hwong@us.ibm.com[2]

***Abstract*—Cognitive radio technology holds great promises in enabling unlicensed operation in licensed bands, to meet the increasing demand for radio spectrum. The new open spectrum operation necessitates novel routing protocols to exploit the available spectrum opportunistically. In this paper we present SAMER, a routing solution for cognitive radio mesh networks. SAMER opportunistically routes traffic across paths with higher spectrum availability and quality via a new routing metric. It balances between long-term route stability and short-term opportunistic performance. SAMER builds a runtime forwarding mesh that is updated periodically and offers a set of candidate routes to the destination. The actual forwarding path opportunistically adapts to the dynamic spectrum conditions and exploits the link with the highest spectrum availability at the time. We evaluate SAMER through both analysis and simulations, and show that it effectively exploits the available network spectrum and results in higher end-to-end performance.**

***Index Terms*—Cognitive Radio, Spectrum Aware Routing**

I. Introduction

The pervasive adoption of wireless services (wireless LAN, wireless mesh networks, Bluetooth) that operate in unlicensed bands such as the 2.4 GHz and the 5 GHz ISM bands, has increased the demand for new spectral resources and more flexible and efficient use of spectrum. Meanwhile, according to Federal Communications Commission (FCC) ([28]), there are intense temporal and geographical variations in the utilization of the licensed spectrum, range from 15% to 85%. For example, the average utilization of the licensed spectrum (TV) broadcast was as low as 14% in 2004 [26]. What is clearly needed is an extension of the unlicensed usage to licensed spectral bands, while accommodating the present users who have legal rights to use this spectrum. As the first step, Federal Communications Commission (FCC) has already announced a new policy of regulating the frequency allocation which allows unlicensed operation in the so-called "white spaces", i.e., spectrum blocks/slices not actively being used by the licensed operators (e.g., TV broadcasters), within the TV broadcast band [26], [29]. To meet the new requirements, Cognitive Radio Technology has been developed which is able to sense the spectral environment over a wide available band and use the spectrum only if communication does not interfere with licensed (*primary*) users. Thus, in Cognitive Radio Networks (CORNETs) the unlicensed low priority (*secondary*) users will be using cognitive radio techniques, to ensure non-interfering co-existence with higher priority users and thus reduce concerns of a general allocation to unlicensed use.

Research on cognitive radio networks has mainly focused on spectrum sensing, management and sharing functionalities which are handled by PHY and MAC layers. IEEE 802.22 [2] is developing a point to multipoint fixed wireless access network standard intended to operate world wide in the unused segments of the terrestrial TV broadcast bands, and it is the first standardization effort to define unlicensed operation in the TV spectrum. DIMSUMnet [3] and DSAP [4] are two cognitive radio architectures that assume a central controller to lease spectrum to users, while KNOWS [1] is distributed and more attractive architecture for cognitive radio mesh networks. However the problem of routing in CORNET has been largely unexplored. In this paper, we study routing over cognitive radio based, static multihop wireless networks.

There are mainly two new issues with routing in CORNETs. First, the concept of "channelization", which serves as the basis for recently proposed routing metrics over wireless mesh networks (e.g., ETX [16], WCETT [7], CAM [10]), is no longer valid. The radio spectrum is dynamically sensed and sliced based on current availability and utilization. Therefore, there are no static channels any more in CORNETs and the routing metrics defined over each static channel need to be adapted. Second, to handle the dynamic variation in the added dimension of spectrum, routing over CORNETs have to balance between long-term (say, over 10s of seconds time scales) route stability and short-term (say, from 10s to 100s of milliseconds time scales) opportunistic performance. Most existing routing protocol operations over mesh networks do not handle both issues.

In this paper, we propose SAMER (spectrum aware mesh routing), a new routing solution for CORNETs that addresses both above issues. The design of SAMER seeks to utilize available spectrum blocks by routing data traffic over paths with higher spectrum availability. In SAMER, routes with highest spectrum availability are selected as candidates. Therefore, SAMER computes its long-term routing metric based on spectrum availability and is more or less a "least-used spectrum first" routing protocol. Moreover, it tries to balance between long-term route stability and short-term route performance via building a runtime forwarding route mesh. Once a route mesh that offers a few candidate routes is computed, the runtime forwarding path is determined by instantaneous spectrum availability at a local node. This may lead to short-term opportunistic performance gain. Our analysis and simulations confirm the effectiveness of SAMER. We show that, under mild long-term spectrum conditions, SAMER can achieve optimal spectrum aware routing.

The rest of the paper is organized as follows. Section II describes the overall system model for SAMER, and section III elaborates on new issues of CORNET routing. Sections IV and V present the design of optimal spectrum aware routing and SAMER. Section VI provides simulation evaluations while in section VII we discuss issues related to our protocol. Finally, section VIII compares SAMER with the related work and section IX concludes the paper.

II. System Model

In this section we provide an overview of the assumptions for the basic functionality of the underlying PHY and MAC layers.

We consider an underlying distributed cognitive radio architecture where PHY and MAC layer platforms work in concert to provide collaborative spectrum sensing and adaptive management and sharing mechanisms. Using collaborative sensing mechanisms cognitive radio users exchange messages about their local view of the spectrum, and they can more effectively detect the primary users. Adaptive spectrum management and sharing mechanisms dynamically adjust the available spectrum blocks to the unlicensed users.

In a cognitive radio environment, each node individually constructs a spectrum allocation matrix, which captures both the operations of the licensed spectrum users, and the secondary user activities. Figure 1 shows a snapshot of a spectrum allocation matrix. A spectrum block is a slice of the available spectrum determined by the MAC and PHY layers and can be specified as a frequency interval $(f_0, f_0 + \Delta f)$ and a time interval $(t_0, t_0 + \Delta t)$. The bandwidth and the time duration of each spectrum block are tuned according to the perceived contention intensity and the total available spectrum resources.

Local spectrum availability at a node i depends not only on the interference i perceives from the primary users, but also on the number of requests from the secondary users for the allocation of the available spectrum. Two communicating nodes have first to contend for spectrum access. The contention can take place in a control channel in unlicensed bands. The spectrum block that will be used for the packet transmissions, can be decided by a handshake procedure between the sender and the receiver. The spectrum that the communicating nodes are allowed to utilize depends on the spectrum availability as defined above. The reservation can be announced on the control channel to inform neighboring nodes for the spectrum usage. Because the spectrum block that will be used for packet transmission is decided locally according to: 1) available spectrum, 2) instantaneous contention intensity, and 3) user traffic demand, the routing protocol cannot pre-specify the interfaces that will be used across the path from source to destination node.

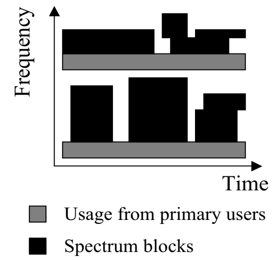


Fig. 1. Spectrum allocation matrix.

A distributed cognitive radio architecture like KNOWS [1] implements the functionalities described above.

III. Routing in Cognitive Radio Mesh Networks

The intense spectrum dynamics of cognitive radio systems, make routing a very challenging and yet unexplored problem. The most challenging issue for a routing protocol in CORNET, is the effective utilization of the available spectrum. In the following section we discuss the limitations of current routing metrics to utilize the available spectrum in CORNET, we introduce the idea of spectrum aware routing and we argue that PHY, MAC and Network layers must work in concert, to achieve optimal routing. From now on, we define optimal routing in terms of 1) hop count (an optimal path must be close in length to the shortest hop-count path), 2) end-to-end throughput and 3) spectrum utilization (an optimal path must exploit all the available spectrum).

A. Why we need a new routing metric?

There are many routing metrics that have been used in single channel wireless multihop networks. Hop-count, per-hop Round Trip Time (RTT) [6], per-hop Packet Pair Delay (PktPair) [15] and Expected Transmission Count (ETX) [16], are some of the most popular metrics. These metrics being designed for single channel networks, cannot be applied without modifications to either dynamic or static spectrum environment.

Multiradio technology offers promising avenue for improving the capacity of multiradio wireless networks ([5], [6]) by enabling nodes to transmit and receive simultaneously through channels that operate on different frequency bands, with different bandwidth, range, and fading characteristics. Single radio metrics have been enhanced to utilize higher spectrum availability, and to encourage channel diversity in order to achieve lower levels of inter/intra path interference. Weighted Cumulative ETT (*WCETT*), Metric of Interference and Channel-switching (*MIC*) and Channel Aware Multipath (*CAM*) metric presented respectively in [7], [9], [10], have been designed for multiradio networks and are based on expected time of a packet transmission (ETT). ETT (presented in [7]) considers: 1) the number of retransmissions required to send unicast packets across a channel by measuring the loss rate of broadcast packets, and 2) the bandwidth of the channel.

As ETT is widely adopted, we use it as a reference. Although these metrics operate in multi-channel environment, they have some limitations when applied in cognitive radio environments for two major reasons::

1) Static multi-channel vs Dynamic spectrum environment: Multiradio metrics have been designed to operate in a static multi-channel environment. In multiradio networks the available radios in each node are fixed during the network deployment. As the channel environment is static, a path is defined as a sequence of interfaces (channels) from a source to a destination node. The multiradio routing approaches address the issues of intra/inter path interference by explicitly defining the sequence of channels to the destination, focusing on encouraging channel diversity. On the other hand in CORNETs, the frequency band that will be used for packets transmission is decided locally according to spectrum conditions, and the routing protocol cannot pre-specify the spectrum blocks that will be used across the path from source to destination. Basically, a path in cognitive radio is defined as a sequence of nodes from source to the destination, while two nodes can be considered neighboring when they have at least one spectrum block in common. As a result, intra/inter path interference cannot be handled explicitly by the routing protocol, but have to be addressed by the underlying protocols. For the same reasons, we cannot pre-compute interference patterns among neighboring nodes. For example MIC metric ([9]) captures inter-path interference by having each node i to keep the set of neighbors ($N_i(c)$) that it interferes with when it transmits on channel c. In multiradio networks the set $N_i(c)$ can be estimated when the network is established, while in CORNETs this is not possible as the spectrum availability changes with time.

2) The concept of spectrum availability: Routing in cognitive radio networks is a two dimensional problem as it has to address: *1)* Spectrum quality, and *2)* Spectrum availability. Spectrum quality refers to different characteristic of a spectrum block such as bandwidth, error rate, and path-loss. Spectrum availability between a pair of nodes (i, j) is determined by two factors: 1) the number of spectrum blocks and the aggregate bandwidth across (i, j), 2) how much of this spectrum is not used by other secondary users. The first factor depends on the interference that nodes i, j perceive from the licensed users. The second factor depends on the traffic load routed through (i, j). The metrics that used both in single radio and in multiradio environments address the spectrum quality and not the spectrum availability dimension. For example, although ETT can be applied in each spectrum block capturing its quality, it does not explicitly consider the impact of contention due to traffic from nearby nodes as it is stated in [7].

In the following section, we show the need of a new spectrum aware routing approach by presenting an example routing scenario.

B. Spectrum aware routing

In this section we introduce the idea of spectrum aware routing by illustrating a simple routing scenario.

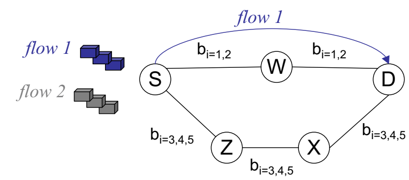


Fig. 2. Spectrum Aware Routing: A motivating example.

Let's consider the network of figure 2, where source node S can reach the destination node D across two non-interfering paths $S-W-D$ and $S-Z-X-D$. We assume that the spectrum blocks b_i available at a node are different. Node W has two spectrum blocks available $b_1 = 8Mbps$ and $b_2 = 10Mbps$, while Z, X have $b_3, b_4, b_5 = 6Mbps$. Source and destination nodes (S, D) have all the spectrum blocks available b_{1-5}.

In the beginning of the scenario we assume that only one flow (flow 1) is routed from S to D across $S-W-D$. The maximum throughput achieved is $8Mbps$. After some time, S initiates also flow 2 towards D. ETT metric will favor path $S-W-D$ for flow 2, as it is shorter in hop count with higher bandwidth spectrum blocks. As a result ETT will lead to unbalanced load distribution as all traffic will go through $S-W-D$ path. The maximum throughput in that case for each flow is $4Mbps$.

The idea behind the spectrum aware routing is that it must adapt to spectrum availability dynamics as defined in the previous section. In the example of figure 2 the aggregate bandwidth between every pair of nodes (the sum of the bandwidth of the available spectrum blocks) is $18Mbps$. When traffic of flow 1 is routed through $S-W-D$, the spectrum availability of this path will be reduced and the spectrum aware routing protocol should send the traffic of flow 2 across $S-Z-X-D$. This results in traffic load distribution between the two different paths, and leads to higher end-to-end performance as for flow 1 throughput will be $8Mbps$ and for flow 2 will be $6Mbps$.

Spectrum aware routing algorithm opportunistically routes data across paths with higher spectrum availability, achieving utilization of all the available spectrum.

C. A cross layer approach

Routing solutions in cognitive radio networks that either completely ignore short-term local spectrum conditions or they are based only on these conditions and do not have any global view of the spectrum, can lead to sub-optimal routing. In this section we argue that an optimal routing solution in CORNET necessitates the collaboration among PHY, MAC and Network layers. A simulation-based comparison between a decoupled route selection and a cross layer routing approach presented in [12], [13], shows a clear benefit in end-to-end performance for cross layer design. We also make our argument clearer by presenting an example.

A routing scenario that illustrates the idea of cross-layer

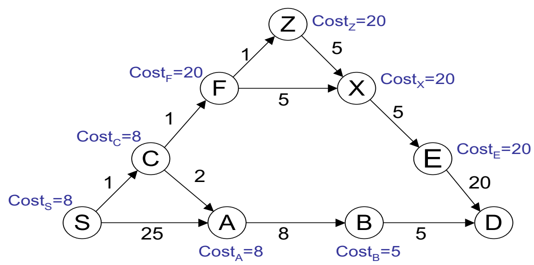


Fig. 3. Cross-layer routing

approach is shown in figure 3. In this scenario nodes implement cognitive radio functionality and the link weights reflect both spectrum availability and quality. Source and destination nodes are S and D respectively. Firstly, let's consider that route selection and spectrum management are decoupled as described in [12] and that each node selects its candidate forwarding node using hop count (the best candidate is the forwarding node across the shortest hop-count path). In case that the hop-count is the same for all the candidate forwarding paths, data is forwarded over the link of the smallest weight. In this routing scenario, in the first step of the algorithm the only candidate forwarding node is A, as C is on a longer path towards D. Then S will route its packets across $S-A-B-D$ towards the destination D ignoring the high cost of its links.

Let's now assume the opposite forwarding approach where routing is handled by the MAC layer and where a node opportunistically forwards data across links with maximum available spectrum and quality (links with low weight). To avoid deviating too much from the shortest hop-count path, the next hop must be in a path which is at most n hops longer from the optimal. In our example (figure 3) we consider $n = 1$. In this case S has two candidate forwarding nodes A, C. The MAC layer is going to forward the packet to C as the $S-C$ link is better. Node C has also two candidate forwarding nodes A, F and is going to pick F for the same reason as before. The final path is going to be $S-C-F-Z-X-E-D$ which is sub-optimal both in number of hops and in spectrum quality and availability.

In cognitive radio routing a cross layer approach must be adopted where spectrum management must work in concert with routing mechanisms.

IV. Optimal Spectrum Aware Routing

The objective of spectrum aware routing is to opportunistically route data packets, across paths with under-utilized spectrum, avoiding congested (in terms of spectrum availability) areas. To achieve this goal, spectrum aware routing must be optimal in distributing traffic according to the available spectrum resources. We formulate this problem as minimization of the spectrum utilization between every pair of nodes (i, j). In this section we will prove that routes which minimize spectrum utilization (optimal routes), can be reproduced as minimum cost (shortest) paths in terms of positive link weigths $w_{i,j}$ which reflect spectrum availability.

We formulate our problem as a Dual Linear Programming problem as presented in [17]. The background of dual linear programs which can be found in [22], [23]. Before we go on with the proof, we introduce some basic notations. Let c_b be the capacity of a spectrum block $b \in B_i$, where B_i is the set of the spectrum blocks available at a node i at time t. The link capacity between a pair of nodes (i, j) can be defined as $C_{ij} = \sum_{b \in B_i \cap B_j} c_b$.

Let T be a traffic matrix where entry $T(s_r, t_r) = d_r$ denotes the average intensity of traffic entering the network at ingress router s_r and exiting at egress router t_r for a commodity $r \in R$. Moreover, X^r_{ij} is the fraction of traffic for commodity r that flows through link (i, j). Spectrum utilization across a link (i, j) can be defined as $u_{ij} = \frac{\sum_{r \in R} d_r X^r_{ij}}{C_{ij}}$ where $\sum_{r \in R} d_r X^r_{ij}$ is the sum over all demands of the amount of flow for that demand which is sent over (i, j).

Cognitive radio network can be modeled as a directed graph $G = (V, E)$ with $v = \| V \|$ mesh routers and $e = \| E \|$ directed links. We formulate the problem of the minimization of spectrum utilization as a linear program (primal LP):

$$min \sum_{(i,j) \in E} u_{ij} \text{ or } min \sum_{(i,j) \in E} \sum_{r \in R} d_r X^r_{ij} \quad (1)$$

subject to

$$\sum_{j:(i,j) \in E} X^r_{ij} - \sum_{j:(j,i) \in E} X^r_{ji} = \begin{cases} 0 & i \neq s_r, t_r (i) \\ 1 & i = s_r, (ii) \\ & r \in R \end{cases} \quad (2)$$

$$\sum_{r \in R} d_r X^r_{ij} \leq C_{ij}, \ (i, j) \in E \ (3)$$
$$0 \leq X^r_{ij} \leq 1, \ (i, j) \in E, \ r \in R \ (4)$$

where X^r_{ij} as we mentioned above is the fraction of traffic for commodity r that flows through link (i, j). The constraints in (2) are flow conservation constraints. The *2(i)* constraint says that the traffic flowing into a node has to equal the traffic flowing out of the node for any node other than source and destination node for each demand. The constraint *2(ii)* basically says that network flow out of the source is 1.

Solving the primal linear program (e.g. using classic Simplex method [23]) we get the optimal solution $\{\bar{X^r_{ij}}\}$ which gives an optimal route or a set of routes (splitting) for each demand. In case that a demand has to be split, it also gives the proportions according to which the traffic between the source and the destination nodes should be distributed across multiple paths.

To prove that optimal routes can be reproduced as minimum cost (or maximum spectrum) paths, we formulate the dual linear program as defined in [17]:

$$max \sum_{r \in R, t \in V} d_r U^r_{t_r} - \sum_{(i,j) \in E} C_{ij} W_{ij} \ (5)$$

subject to

$$U^r_j - U^r_i \leq W_{ij} + 1, \ \forall r \in R, \ (i, j) \in E \ (6)$$
$$W_{ij} \geq 0$$
$$U^r_{s_r} = 0$$

Let the optimal solution for the dual program be $\{\bar{U_i^r}\}$ and $\{\bar{W_{ij}}\}$. If we view $\{\bar{W_{ij}}\}$ as constant, $\{\bar{W_{ij}}+1\}$ can be considered as link weights. For simplicity, we consider $w_{ij} = \{\bar{W_{ij}}+1\}$ as the weight of the link (i, j) where $w_{ij} > 0$ as $\bar{W_{ij}} + 1 \geq 0$. In addition, the optimal solution of the dual program $\{\bar{U_i^r}\}$, can be viewed as the length (the sum of link weights) from source s_r to a node i.

By applying the complementary slackness theorem we get:

Lemma: If P_r is an optimal route determined by the X_{ij}^r values, then for every link $(i, j) \in P_r$ if $X_{ij}^r > 0$, then $\bar{U_j^r} - \bar{U_i^r} = w_{ij}$.

As X_{ij}^r represents the spectrum utilization in terms of traffic routed across (i, j), weights w_{ij} reflect the spectrum availability between a pair of nodes (i, j), so small w_{ij} implies high spectrum availability. In the following theorem we prove that optimal routes derived from the primal linear program can be reproduced as minimum cost paths with respect to w_{ij}. Notice that the cost of a path P in terms of spectrum availability is equal to the maximum w_{ij} for $(i, j) \in P$, as the spectrum availability across a path is determined by the bottleneck link (link with lowest spectrum availability).

Theorem: Let P be a path from s_r to t_r and for every link $(i, j) \in P$, $\bar{U_j^r} - \bar{U_i^r} = w_{ij}$. Then P is the minimum cost path with respect to link weights $\{w_{ij}\}$.

Proof: Firstly let's consider that p_j with $0 < j < n$ are the nodes of the path $P = p_0, p_1, ...p_{n-1}, p_n$ where $p_0 = s_r$ and $p_n = t_r$. Then we have

$$\bar{U_{p_j}^r} - \bar{U_{p_{j-1}}^r} = w_{p_{j-1}p_j}$$

for $0 < j < n$.

As we mentioned above, in spectrum aware routing a weight $w_{p_{j-1}p_j}$ for $0 < j < n$ reflects the spectrum availability, and the cost of the path is determined by the bottleneck link, the link with maximum $w_{p_{j-1}p_j}$. So the cost of a path P is defined as:

$$C_{t_k}^P = max\{w_{p_{j-1}p_j}\}_{0<j<n}$$

Let now consider another path $Z = z_0, z_1, ...z_{m-1}, z_m$ where $z_0 = s_r$ and $z_m = t_r$. From constraint (6):

$$\bar{U_{z_j}^r} - \bar{U_{z_{j-1}}^r} \leq w_{z_{j-1}z_j}$$

which implies that $C_{t_k}^P \leq C_{t_k}^Z$. As a result P is the minimum cost path or the maximum spectrum path.

■

We proved that optimal routes which achieve minimum spectrum utilization in cognitive radio mesh networks can be reduced into maximum spectrum paths with respect to a set of positive link weights. These link weights reflect spectrum availability.

Optimal minimum spectrum utilization can be obtained theoretically by solving a linear program, however this theoretical approach is not easy to be applied in mesh networks. Formulating the linear program, requires centralized knowledge of the traffic demands between each source and destination pair. However traffic demands change very frequently and are difficult to acquire. Moreover, the solution of the linear program requires the ability to split traffic arbitrarily among all paths between a source and a destination, which is hard to achieve in reality since it introduces high complexity into the routing mechanism and may also cause out-of-order delivery of TCP traffic [21]. As these optimal routes are reduced to positive link weights which reflect spectrum, heuristics that capture spectrum availability must be designed.

V. SAMER DESIGN

In this section we present SAMER, a routing protocol for cognitive radio mesh networks, whose goal is to opportunistically utilize the spectrum in the network, by routing traffic across paths with higher spectrum availability while at the same time it achieves long-term stability by not deviating from the shortest hop-count path. We show that by exploiting all the available spectrum, SAMER can achieve eventually high end-to-end performance.

SAMER builds a forwarding mesh which is adjusted periodically according to the spectrum dynamics, and opportunistically routes packets across this mesh. The mesh is centered around the long-term shortest path (in terms of hop-count), but opportunistically expands or shrinks periodically to exploit spectrum availability. In short, SAMER takes a two-tier routing approach and balances between long-term optimality (in terms of hop count) and shortest opportunistic gain (in terms of higher spectrum availability). SAMER has main two components:

- *Dynamic Candidate Mesh*: Every node in the network computes a cost to the destination D (for each destination each node computes a different cost). This cost reflects the spectrum availability of the highest spectrum path whose length is less than H hops. Also every node builds a set of candidate forwarding nodes to D, by including all its neighboring nodes whose cost to D is less than a threshold C . So the mesh is built around the shortest in hop count path and is dynamically adapted to spectrum changes.
- *Opportunistic Forwarding*: SAMER opportunistically forwards packets across the links with the highest spectrum availability. Upon a reception of a packet a forwarding node chooses from the links included in the candidate set, the one with the highest spectrum availability. For computing spectrum availability we use PSA metric as defined in section V-B.

SAMER succeeds in balancing between long-term stability as the paths to the destination do not divert much from the shortest path, and short-term opportunistic utilization of the spectrum. In the following section we present the building blocks of the dynamic candidate mesh.

A. Building a candidate forwarding mesh

SAMER builds a forwarding mesh around the long-term shortest path and adjusts it periodically according to spectrum dynamics. Using this mesh, it greedily forwards data packet across the link with the highest spectrum availability. The forwarding mesh is built by computing for each node i a cost $Cost_i$. In the simplest case $Cost_i$ reflects the spectrum availability of the highest spectrum path whose length is less than H hops. So it computes all the paths of at most H hops and from these it selects the one with the highest spectrum availability (if there is no such path cost is set to infinity). The appropriate value for H is a difficult decision. By setting a small value for H we may not discover all the paths to the destination or we may not discover the paths with low $Cost_i$. A more flexible cost computation algorithm that can adapt to different application requirements is discussed in section V-C2. The algorithm increases H until it finds at node i a cost where $Cost_i \leq C_{max}$. C_{max} can be considered the maximum allowable cost to the destination. This is a double objective optimization problem as we would like to minimize H while at the same time maximize the spectrum availability (maximize spectrum availability reflects in minimizing the cost). In section V-C2 we solve this problem using distributed Bellman Ford.

By considering hop-count, we achieve long-term stability, as all the candidate paths towards the destination, are centered around the shortest hop-count path. Except from stability, shorter hop-count paths consume a minimal amount of network resources. By considering spectrum availability, we have a global view of spectrum dynamics. In section III-C we present examples that illustrate that routing based only on local view of spectrum availability can lead to congested links. However we can limit these problems if we have a global view of the spectrum. We realize, that it is very important for each node to have an updated global view of the spectrum availability.

To sum up, in each forwarding step when a node i forwards a packet Pkt towards a destination node D, it performs the following actions:

- *Action 1*: Node i computes $Cost_n$ for $\forall n \in N$ where N is the set neighboring nodes of i. Node i executes a link state (e.g. OSPF) protocol so it has all the information to compute $Cost_n$ for $\forall n \in N$.
- *Action 2*: Node i adds to its forwarding candidate set $Candidate_i$ all the nodes n where $Cost_n \leq C$.
- *Action 3*: Node i forwards Pkt to the highest spectrum availability link (i, n) where $n \in Candidate_i$. Spectrum availability is computed using PSA metric described in the following section.

In this paper we consider that cost C is determined by each node separately and it is independent of the flow. The value of C is a trade-off between long and short term performance. So if C value is high, we focus more on short term path properties. This will be the recommended approach if the spectrum dynamics are very intense while the periodic updates about the spectrum availability is not very frequent which means we have an outdated global view of the spectrum. On the other hand, the value of hop count H determines how much the algorithm expands or shrinks the forwarding mesh.

In the following sections we will present PSA, a metric for estimating path spectrum availability and the algorithm that we use to compute the cost at a node i ($Cost_i$). Finally we explore how SAMER works in the routing scenario presented in section III-C.

B. PSA metric

Optimal spectrum utilization in cognitive radio mesh networks can be achieved by assigning positive link weights which reflect spectrum availability. Intuitively this can be succeeded by routing traffic across paths with the most under-utilized spectrum.

In this section we present the basic component of SAMER which is Path Spectrum Availability (PSA) metric. PSA metric is used to favor paths with higher spectrum availability and quality. By exploiting under-utilized spectrum, PSA can lead to higher end-to-end performance. To capture spectrum availability and quality PSA metric considers: 1) *Local spectrum availability*: Spectrum availability at a node i depends on a) the number of available spectrum blocks at i and their aggregated bandwidth and b) on how much of this spectrum is not allocated from secondary users. 2) *Spectrum block's quality*: The quality of the spectrum block refers to its bandwidth and loss rate. Loss rate depends both on each frequency band's properties and the interference it perceives from both primary and secondary users.

To compute PSA metric we first calculate the probability p_{loss} that a packet transmission between a pair of nodes (i, j) is not successful. The 802.11 protocol considers that a transmission is successful, if the packet is also successfully acknowledged. So we will also consider the packet loss probability in both the forward and reverse directions; these probabilities are denoted p_f and p_r respectively. Loss probability p_{loss} is: $p_{loss} = 1-(1-p_f)\times(1-p_r)$. Moreover, p_{loss} can be estimated by measuring the loss rate of broadcast packets between pairs of neighboring nodes as proposed in [16].

Let's also define $T_{f,b}$ as the fraction of time during which the node i is free to transmit and/or receive packets through a spectrum block b. Respectively, $T_{w,b}$ is the fraction of time that the node i has to wait as b is busy. The fraction of time T_w includes: *1)* T_a is the fraction of time during which a node i is receiving/transmitting packets for successfully received/transmitted packets, *2)* T_i is the fraction of time during which a node i is deferring for transmissions that interfere with it, *3)* T_c is the fraction of time during which a node i experienced packet collisions. So for every spectrum block b, $T_f = 1-T_w = 1-(T_a+T_i+T_c)$. All of the required information (T_a, T_i, T_c) can be derived from the MAC layer, and even though most commonly available device drivers do not export interfaces to higher layers such as the routing layer for extracting these values, similar information is exposed by some wireless cards such as the DARPA GloMo Radio API.

The fraction of time $T_{f,b,i}$ that two neighboring nodes (i, j) are free to transmit/receive across a spectrum block b, can be different as (i, j) may perceive different interference signals both from licensed users and from their neighboring unlicensed users. However in many cognitive radio applications (cognitive radio networks in TV broadcast bands) the transmission power of licensed users is very intense and their signals provide coverage areas with much greater radius than the transmission capability of unlicensed wireless device. So the availability of a block b (the time that b is not allocated by the licensed users) between two neighboring nodes (i, j) can be considered equal. Although the availability of b can be considered equal, the contention from the unlicensed nodes in range is different, so we consider that $T_{f,b} = min\{T_{f,b,i}, T_{f,b,j}\}$ (this information can be derived from the MAC layer protocol). The throughput that can be achieved between a pair of nodes (i, j) across a spectrum block b will be formulated as :

$$Thr_{(i,j),b} = T_{f,b} \times B_{w,b} \times (1 - p_{loss,b})$$

$B_{w,b}$ is the bandwidth and $p_{loss,b}$ the loss probability of the spectrum block b.

The aggregate throughput between a pair of neighboring nodes (i, j) is given by:

$$Thr_{(i,j)} = \sum_{b \in B_i \cap B_j} a_b \times \frac{Thr_{(i,j),b}}{maxThr_{(i,j),b}}$$

where $maxThr_{(i,j),b}$ is the $Thr_{(i,j),b}$ when $p_{loss,b} = 0$ and $T_{f,b} = 1$ and B_i are the spectrum blocks available at a node i. In addition each spectrum block b is assigned a weight a_b where $a_b \leq 1$ for every spectrum block. This weight is determined by the physical characteristics of b. Many different parameters define the quality of each spectrum band such as the interference level, channel error rate, path-loss, link layer delay. Moreover two neighboring blocks in terms of frequency may be not completely non-interfering. The weight a_b can be used to capture all these different spectrum properties. We will explore these characteristics in future work. So from now on we will consider $a_b = 1$ for every spectrum block b.

The *Smoothed Aggregate Throughput* between a pair of neighboring nodes (i, j) is defined as:

$$SThr_{(i,j)} = \alpha \times SThr_{(i,j)} + (1 - \alpha) \times Thr_{(i,j)}$$

For a pair of neighboring nodes (i, j) the $SThr_{(i,j)}$ is the smoothed aggregate throughput which reflects both our current view and statistical information about spectrum availability. For our simulations we consider $a = 0.4$.

The Smoothed Aggregate Throughput can be assigned as a positive link weight. Path spectrum availability across a path P is defined as

$$PSA_P = min\{SThr_{(i,j)}\}_{(i,j) \in P}$$

Spectrum availability for a path P is considered the minimum Smoothed Aggregate Throughput for $(i, j) \in P$ as this link is going to be the bottleneck which affects the whole path.

C. *Cost computation*

In SAMER each node has a full topology map and it can compute a cost $Cost_i$ $\forall i \in N$ where N is the set of nodes, to every destination node D. In the simplest approach, $Cost_i$ reflects the spectrum availability of the highest spectrum path P whose length is less than H hops. Spectrum availability across a path P can be computed using the metric PSA_P. So if between every pair of nodes (i, j) we assign a weight $SThr_{(i,j)}$, the spectrum availability for a path P will be $PSA_P = min\{SThr_{(i,j)}\}_{(i,j) \in P}$ according to the definition in the previous section.

In this section we describe cost computation as double objective optimization problem where we increase H until we find at node i a cost where $Cost_i \leq C_{max}$. In the following paragraphs we discuss the advertisement of link state information $SThr_{(i,j)}$, the cost computation algorithm and the value of C_{max}.

1) Advertisement of link state information: The Smoothed Aggregate Throughput $SThr_{(i,j)}$ assigned between every pair of nodes (i, j) need to be advertised across the network, so that each mesh router can compute PSA_P for a path P. As energy consumption and limited processing power are not an issue in wireless mesh networks we consider that SAMER uses an extended OSPF mechanism for link state advertisement, where each mesh router maintains an updated database of the network topology, including the current state ($SThr_{(i,j)}$) of each link. To have a consistent global view of spectrum dynamics, every node i must update PSA_P (and as a result its $Cost_i$) in short time periods. The frequency of OSPF LSA updates, is based on a tolerable corresponding load on the network. A different approach is, instead of sending periodic link state updates we can trigger link state advertisements only when there is a significant change in the value of $SThr_{(i,j)}$ since the last advertisement. We plan to compare these options in future work.

2) Cost computation algorithm: Cost computation problem is a double objective optimization problem as we would like to minimize H while at the same time maximize the spectrum availability (maximize spectrum availability reflects in minimizing the cost). Nevertheless because of the specific nature of the two objectives being optimized, the complexity of the above algorithm is competitive with even that of standard single-objective algorithms.The Bellman-Ford (BF) shortest path algorithm is the excellent candidate for cost computation as it can be easily adapted to compute paths of maximum available spectrum for all hop counts. It is a property of the BF algorithm that, at its $h - th$ iteration, it identifies the optimal (maximum spectrum availability) path between the source and each destination, among paths of at most h hops. Specifically, at the kth (hop count) iteration of the algorithm, the maximum spectrum available to all destinations on a path of no more than k hops is recorded (together with the corresponding routing information). The algorithm terminates, when it provides for all destinations, the path with the smallest possible number of hops which satisfy the constraint $Cost \leq C_{max}$. This path is also the one with the highest spectrum availability among all the paths with at most these many hops. This is because for any hop count, the algorithm always selects the one with maximum spectrum availability. The pseudocode and the data structures of the algorithm are presented in the appendix.

A crucial issue is when to invoke the cost computation algorithm. One approach is to trigger a computation for each new request for packet transmission which however can be very computationally expensive, In our evaluation we adopt the pre-computation approach where costs are pre-computed to all destinations periodically. As a future work we will explore how SAMER will perform if cost information is more frequently updated.

3) The parameter C_{max}: The maximum allowable cost to the destination C_{max} has the same meaning as the upper bound of the cost used to select candidate neighbors (value C in section V-A) and it is a tradeoff between short and long term performance. So C_{max} should be set equal to C. If $C_{max} > C$ the node may not be able to find candidate nodes to further forward the data. Based on the C_{max}, C values we can easily enchance our solution to guarantee robust data delivery and support for differentiated services (see [24]). The study of these mechanisms is out of the scope of this paper.

D. SAMER in an example routing scenario

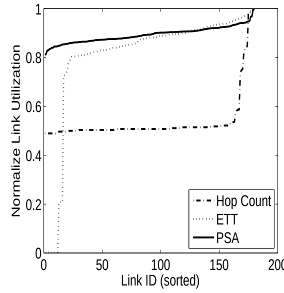
In routing scenario presented in section III-C (figure 3), we showed that routing approaches that decouple Network and MAC layer can lead to sub-optimal routing in cognitive radio mesh networks. Let's consider that in the topology of figure 3, the mesh nodes execute SAMER routing protocol where H is equal to the diameter of the network. In the example topology except from the link weights, we present also the cost $Cost_i$ for each node i and we also consider for every node $C = 8$. In the first round, the algorithm has two candidate forwarding nodes C, A, and because link $S - C$ is better, it will forward the packet to C. In the second round, C has only one candidate, node A as $Cost_F > 8$. So the final path to the destination D, will be $S - C - A - B - D$ which is one hop longer than the shortest path and it has the highest spectrum availability.

VI. Evaluation

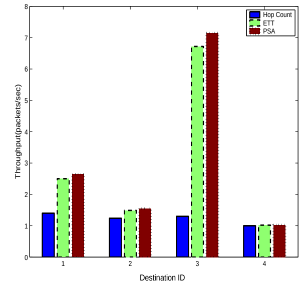
In this section we evaluate the effectiveness of SAMER to 1) distribute data traffic across paths with under-utilized spectrum, and 2) improve end-to-end throughput.

A. Simulation setup

We perform our simulations using Qualnet [18]. We randomly deploy 52 static equivalent (with the same radio capabilities) nodes in a $1500m \times 1500m$ terrain. The available frequency band has aggregated bandwidth 10Mbps and can be divided by the cognitive radios in 5 spectrum blocks of 2Mbps each. For simplicity, we assume that these spectrum blocks have the same packet loss ratio. A node i can have from 0 to 5 spectrum blocks b available at a time t. A spectrum block b is considered available if it is not allocated from a neighboring node or a primary user. Two neighboring nodes can exchange data if they have at least one common block available. Each time t a spectrum block at a node i is allocated by a primary user with probability P (we set $P = 0.1$). As it is stated in section V-B this assumption is not true in some applications as in CORNET in TV broadcast bands.



(a) Link Utilization



(b) TCP Throughput

Fig. 4. Link utilization and throughput performance of 3 routing metrics

Metrics	Min.	Max.	Mean	StdDev
Hop count	0.489	1.0	0.537	0.102
ETT	0.001	1.0	0.807	0.243
PSA	0.811	1.0	0.893	0.032

TABLE I
MIN., MAX., MEAN AND STANDARD DEVIATION OF NORMALIZED LINK UTILIZATION OF THE 3 DIFFERENT ROUTING METRICS

The SAMER was simulated as described in V. We use OSPF [25] to advertise link state information periodically, while we use the pre-computation approach to compute the costs at each node. For our experiments we set $C^{id}_{max} = C^{id}_{Aver}$ from a node i to a destination node d where C^{id}_{Aver} is the maximum cost path from i to d divided by two ($C^{id}_{Aver} = max\{Cost^{id}\}/2$). Intuitively we pick this value to balance between short and long term performance. The study of how C_{max} affects SAMER's performance is a part of future work.

In the following sections we compare the performance of SAMER using PSA, ETT metric on each spectrum block with the shortest path (in terms of hop-count) distributed Bellman Ford algorithm. To evaluate the effectiveness of these approaches, we create intense TCP traffic that is destined to four nodes located at the edge of the mesh topology. The following results are the average of many experiments.

B. Link utilization

We first evaluate the effectiveness of each metric to distribute the traffic load across different paths to achieve load balancing. In table I, we present the Minimum, Maximum, Mean, and Standard Deviation of normalized link utilization, while in figure 4a we illustrate the normalized utilization for every link in the network.

From table I we observe that the standard deviation of the link utilization for PSA metric is an order of magnitude smaller than hop count and ETT metric; the difference between minimum and maximum link utilization of PSA is much smaller than the other two metrics. This result shows that PSA can effectively distribute data traffic across different paths. One interesting observation is that the ETT performs worse than the others metrics in balancing traffic load across different paths. The reason is that ETT favors high quality paths and it does not explicitly consider the impact of contention due to traffic from nearby nodes as stated in [7].

Figure 4a shows a detail view of the link utilization for each link. The sharp curves for ETT and hop-count metric reveal that many links are either over-utilized or under-utilized. On the other hand, the PSA metric results in a relatively flat distribution, which means load is balanced across the network.

C. Throughput gain

In this section we study the end-to-end network throughput, and we present our results in figure 4b. Each bar in figure 4b represents the throughput for PSA, ETT and Hop-count in log scale. In general, PSA and ETT metrics perform better than hop count. This is because the hop-count approach considers neither spectrum availability nor spectrum quality. Hop-count metric does not consider the spectrum dynamics which are resulted from the interference of the primary users and which lead to changes in bandwidth availability. This results in low end-to-end throughput. On the other hand, both PSA and ETT metrics consider the path quality and choose high bandwidth paths. PSA metrics is slightly better (about 0.5% to 6.4%) than the ETT metric as it also explicitly considers spectrum availability in terms of traffic load, so it avoids congested links.

Finally, we observe that for the destination node 4, and the three metrics seem to achieve almost the same throughput performance. After examining the routes of the TCP packets, the shortest path (in terms of hop-count) is not so congested as node 4 is at the edge of the network and it is not greatly affected by the cross traffic.

1) Throughput gain vs Nodes Density: We carry another set of simulation to study the effect of varying the node density on throughput. In this simulation, the simulation setting is the same as described above (we use SAMER with PSA), except we increase the number of nodes progressively.

In figure 5 we present for each of the four destination nodes how throughput (in log scale) is changing with the network's density. In all cases we observe a linear increase in throughput until the node density becomes about 60. Then the throughput remains almost the same. The reason is that an increase in nodes density results in an increase in spectrum availability. However after some point, bandwidth is adequate to serve the traffic demands.

In conclusion, PSA is more effective than the other two metrics in distributing traffic across high spectrum availability paths and in balancing utilization among links. This eventually leads to higher end-to-end performance.

VII. Discussion

In the derivation of PSA we do not explicitly consider inter/intra path interference. As we state in section II, in cognitive radio networks the routing protocol cannot pre-specify the interfaces that will be used across the path from source to destination node, as the spectrum block that will be used for packet transmission is decided locally according to 1) available spectrum, 2) contention intensity, and 3) user traffic demand. As a result, a routing metric cannot explicitily address these issues. However we argue that though implicitily, PSA can effectively capture interference. By routing traffic flow across paths with higher spectrum availability we increase the probability that a node across the path will find a spectrum block available to serve the flow.

In our simulations we set in PSA the weight of each spectrum block as $a_b = 1$. The weight a_b captures the quality spectrum block and can be set by the MAC layer. However it also captures the interaction among neighboring spectrum blocks (two neighboring spectrum blocks may be not completely non-interfering). We are planning to explore how the spectrum blocks' characteristics affect spectrum availability in future work.

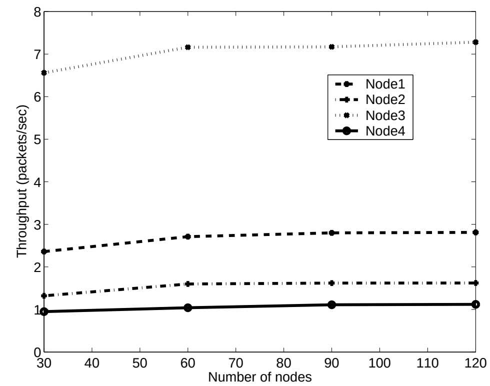


Fig. 5. SAMER's throughput with node's density for each of the destination nodes.

VIII. Related Work

Existing research on cognitive radios mainly focuses on MAC and Physical layer issues. Various spectrum sensing and management solutions have appeared in the literature [3], [4], [12], [14], [19]; a nice survey of this topic is available in [27]. New MAC solutions and prototype systems are also available (e.g., [1] and the references there). Initial studies that examine the interdependence between route selection and spectrum management have also been described [12], [13]. However, their approaches are very different from ours. [12] effectively proposes a decoupled method for route selection and spectrum management. Route selection is still following the shortest path algorithm. Instead, SAMER renovates both the routing metric and routing protocol operations. The focus of [13] is on comparisons of layered and cross-layered approaches. SAMER though proposes a new routing solution over a CORNET.

There have been extensive studies on single-channel and multi-channel wireless mesh networks [7], [9], [10], [11]. One of the main contributions of such early work is on devising new routing metrics that offer improved performance over mesh networks based on conventional radios. Such metrics include Weighted Cumulative ETT (*WCETT*) [7], Metric of Interference and Channel-switching (MIC) [9], Channel Aware Multipath (*CAM*) [10], and Adjusted Expected Transfer Delay [11]. However, such metrics cannot be directly applicable to CORNET. They still measure link quality over well-defined, static channels, and do not take into account of spectrum dynamics which are present in CORNET.

IX. Conclusion

Cognitive radios (CRs) hold promises in significantly increasing radio spectrum utilization through dynamic spectrum sensing and opportunistic utilization. In order to fully exploit the new capability offered by cognitive radio technology at the physical-layer, a cognitive radio network (CORNET) must address the new issues brought out by CR possibly across the entire protocol stack. The fundamental problem is that, resource in a wireless CORNET is no longer a 2-D factor in time and space, but is now a 3-D factor across frequency, time and space. Therefore, protocol design has to explicitly handle the implications brought out by the added dimension of radio spectrum, in order to fully realize the potential of CR nodes in a CORNET. From this perspective, protocol design for CORNET defines another problem domain for cross-layer designs.

In this paper, we propose SAMER, arguably the first routing protocol in the literature that addresses two new routing issues in a CORNET: dynamic spectrum availability without the conventional concept of "channelization" in traditional mesh networks, and tradeoffs between long-term route stability and short-term opportunistic routing performance. To address both issues, SAMER renovates both the routing metric and the routing protocol operations. The routing metric of SAMER explicitly considers both route quality based and high spectrum availability. The ultimate goal is to provide optimal spectrum aware routing in the long term. To increase short-term routing performance, SAMER selects a fine-time-scale (say, 10s to 100s of milliseconds), opportunistic forwarding path out of a mesh of candidate routes computed based on coarse-time-scale (say, 10s of seconds) spectrum availability. This way, SAMER flexibly balances between long-term stable routing and short-term opportunistic forwarding. It serves as another concrete example of applying cross-layer design over a CORNET.

Our analysis and preliminary simulation evaluations have shown SAMER as a viable routing solution that can provide better performance in CORNET. Ongoing work seeks to provide more thorough evaluations via comprehensive simulations and further refine the design of SAMER.

References

[1] Y. Yuan, P. Bahl, R. Chandra, P. Chou, I. Farrel, T. Moscibroda, S. Narlanka, and Y. Wu, *KNOWS: Cognitive Networking Over White Spaces*, in IEEE DySpan 2007.

[2] IEEE 802.22 WRAN WG, *www.ieee802.org/22/*

[3] M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, J. Evans *DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access.* In WOWMOM 2005

[4] V. Brik, E. Rozner, S. Banerjee, and P. Bahl, *DSAP: A Protocol for Coordinated Spectrum Access*, in IEEE Dyspan 2005

[5] P. Bahl, A. Adya, J. Padhye and A. Walman *Reconsidering wireless systems with multiple radios*, SIGCOMM Comput. Commun. Rev., 34(5):39-46, 2004.

[6] A. Adya, P. Bahl, J. Padhye, A. Wolman and L. Zhou *A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks* In BROADNETS '04: Proceedings of the First International Conference on Broadband Networks (BROADNETS 2004).

[7] R. Draves, J. Padhye and B. Zill, *Routing in multi-radio, multi-hop wireless mesh networks*, In MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking, 2004.

[8] R. Draves, J. Padhye and B. Zill, *Comparison of routing metrics for static multi-hop wireless networks* In SIGCOMM '04.

[9] Y. Yang, J. Wang, and R. Kravets, *Interference-aware load balancing for multihop wireless networks*, UIUC Technical report, 2005.

[10] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, *A multi-radio unification protocol for ieee 802.11 wireless networks*. In BROADNETS '04: Proceedings of the First International Conference on Broadband Networks (BROADNETS '04).

[11] W. Zhou, D. Zhang and D. Qiao, *Comparative study of routing metrics for multi-radio multi-channel wireless networks* In Wireless Communications and Networking Conference, 2006. WCNC 2006.

[12] Q. Wang and H. Zheng, *Route and Spectrum Selection in Dynamic Spectrum Networks*, In IEEE Consumer Communications and Networking Conference (CNCC) 2006.

[13] C. Xin, B. Xie and C. Shen, *A novel layered graph model for topology formation and routing in dynamic spectrum access networks*, In IEEE DySPAN 2005.

[14] J. Zhao, H. Zheng and G. Yang, *Distributed Coordination in Dynamic Spectrum Allocation Networks* In IEEE International Symposium on Dynamic Spectrum Access Networks, 2005.

[15] S. Keshav, *A Control - Theoretic Approach to Flow Control*, In Proceedings of the conference on Communications architecture and protocols pages 3-15, 1993.

[16] D. Couto, D. Aguayo, J. Bicket and R. Morris, *A High-Throughput Path Metric for Multi-Hop Wireless Routing* In MOBICOM 2003.

[17] Y. Wang, Z. Wang and L. Zhang, *Internet Traffic Engineering without Full Mesh Overlaying.* In INFOCOM 2001.

[18] Scalable Network Technologies, *QualNet Simulator*

[19] R. W. Brodersen, A. Wolisz, D. Cabric, S. M. Mishra, and D. Willkomm, *Corvus: a cognitive radio approach for usage of virtual unlicensed spectrum*,in DySPAN 2005.

[20] S. Haykin, *Cognitive radio: brain-empowered wireless communications*, IEEE Journal on Selected Areas in Communications 23 (2) (2005).

[21] J. Wang, Y. Yang, L. Xiao, and K. Nahrstedt, *Edge-based Traffic Engineering for OSPF Networks* Elsevier Journal on Computer Networks, vol. 48, no. 4, 2005.

[22] G. Nemhauserand L. Wolsey, *Integer and Combinatorial Optimization*, WILEY, 1988

[23] V. Chvatal, *Linear Programming*, W H Freeman and Company, 1983.

[24] Y. Yuan, S. H.Y. Wong, S. Lu, and W. Arbuagh *ROMER: Resilient Opportunistic Mesh Routing for Wireless Mesh Networks* In IEEE WiMesh 2005

[25] J. Moy, *OSPF Version 2*, RFC 2328. Apr 1998.

[26] FCC *Unlicensed operation in the tv broadcast bands, notice of proposed rulemaking*, ET Docket No. 04-186, May, 2004.

[27] I.F. Akyildiz, W-Y Lee, M.C. Vuran, and S. Mohanty, *Next Generation/dynamic spectrum access/cognitive radio wireless networks: A Survey*, in Computer Networks 50, 2006.

[28] FCC, *Notice of proposed rule making and order*,ET Docket No 03-222, December 2003.

[29] Office of Engineering and Technology. *Projected Schedule for Proceeding on Unlicensed Operation in the TV Broadcast Bands*. In Sep., 2006.

Appendix

A. Pseudocode for the Bellman Ford Pre-Computation Algorithm

The pseudocode presented below, assumes a hop-by-hop forwarding approach in updating the neighbor field. Moreover it does not handle equal cost multi-paths for simplicity, but the modification needed to support this functionality is straightforward. Finally in the pseudocode we use the term ”interface” to define the next hop in the path towards the destination.

Algorithm 1 Bellman Ford Pre-Computation Algorithm

Input:
V = Set of vertices, labeled by integers 1 to N.
L = Set of edges, labeled by ordered pairs (n,m) of vertex labels.
S = Source vertex (which executes the algorithm).

for all edges (n,m) in L: **do**
  $sthr(n,m)$ = Smoothed aggregate throughput (according to last received update) on interface associated with the edge between vertices n and m.
  $iface(n,m)$ = Outgoing interface corresponding to edge (n,m) when n is a router.
  H = Maximum hop-count (at most the graph diameter).
end for

Type:
$rtable\_entry$: Routing table record with two entries:
 i) psa = integer,
 ii) $neighbor$ = integer 1..N.

Variables:
$RT[1..N,1..H]$: Routing table, whose (n,h) entry is a rtable_entry record, such that:
 i) $RT[n,h].psa$ is the path spectrum availability bandwidth (as known so far) on a path of at most h hops between vertices s and n,
 ii) $RT[n,h].neighbor$ is the first hop on that path (a neighbor of s). It is either a router or the destination n.
Set_prev: list of vertices that changed a psa value in the RT table in the previous iteration.
Set_new: list of vertices that changed a psa value in the RT table in the current iteration.

The Algorithm:
begin:
for n:=1 to N **do**
  /* initialization */
  $RT[n,1].psa := 0;$
  $RT[n,1].neighbor := null;$
end for
$RT[s,1].psa := infinity;$
$reset\ Set\_prev;$
for all neighbors n of s **do**
  $RT[n,1].psa := max(RT[n,1].psa, sthr[s,n]);$
  if RT[n,1].psa = sthr[s,n] **then**
    $RT[n,1].neighbor := iface(s,n);$
  end if
  $S\_prev := S\_prev\ union\ \{n\};$
end for

for h:=2 to H **do**
  $reset\ Set\_new;$
  for all vertices m in V **do**
    $RT[m,h].psa := RT[m,h-1].psa;$
    $RT[m,h].neighbor := RT[m,h-1].neighbor;$
  end for
  for all vertices n in Set_prev **do**
    for all edges (n,m) in L **do**
      if $min(RT[n,h-1].psa, sthr[n,m])$
      $> RT[m,h].psa$ **then**
        $RT[m,h].psa$
        $:= min(RT[n,h-1].psa, sthr[n,m]);$
        $RT[m,h].neighbor$
        $:= RT[n,h-1].neighbor;$
        $Set\_new := Set\_new\ union\ \{m\};$
      end if
    end for
  end for
  $S\_prev := S\_new;$
  if S_prev=null **then**
    $h = H + 1;$
    /* if there are no changes, exit */
  end if
end for
end: