Channel Dynamics Matter: Forwarding Node Set Selection in Cognitive Radio Networks

Ying Dai, Jie Wu, and Andrew Daniels

Department of Computer and Information Sciences, Temple University
Motivation

- Routing in Cognitive Radio Networks (CRNs): Primary Users + Secondary Users
- Reliability becomes the main concern.
- Opportunistic routing is a promising solution.
  - Nodes do not stick to a particular route.
**Problem**

- Question: Are the previous works on opportunistic routing directly applicable on CRNs?
Question: Are the previous works on opportunistic routing directly applicable on CRNs?

The answer is negative.

One problem is the forwarding node set selection.
  - PUs have to be taken into consideration.
Problem

- Activities of primary users in CRNs are unknown and unpredictable.
- An optimal solution is unrealistic in such a dynamic environment.
- Challenge: How to select the forwarding node set in CRNs?
To select forwarding sets for each sender, the neighbor nodes need to be prioritized first.

After defining the weights for each neighbor, the basic algorithm, the greedy algorithm with one backtrack scheme, and the maximum weighted independent set algorithm are proposed.
Forwarding Node Prioritization

- Two factors need to be considered:
  - Interference from primary users;
  - Interference from other nodes.
Two factors need to be considered:

- Interference from primary users;
- Interference from other nodes.
Forwarding Node Prioritization

- We use a conflict probability of adjacent links to a link $uv$ to help define the weight of a forwarding node:

\[
C_{uv}(m) = \sum_{w \in N_v} \frac{1}{|M_{uw}|} E_{uw}(m) + \sum_{w \in N_u} \frac{1}{|M_{vw}|} E_{vw}(m),
\]

where $N_v$ is the set of neighbor nodes for $v$; $M_{uv}$ denotes the channels available on link $uv$, and $M_{uv} = M_u \cap M_v$; $E_{uw}(m)$ is a step function with a value of 1 when a channel $m \in m_{uw}$ and is 0 otherwise.
Based on the definition of $C_{uv}(m)$, we now define the receiving ability of node $v$ from its sender $u$:

$$R_{uv} = \sum_{m \in M_{uv}} \frac{1}{C_{uv}(m)}.$$

Obviously, node $v$, with more channels that are less likely to be used by adjacent links, has a better value of $R_{uv}$. 
Finally, for $\forall v \in N_u$, the weight of $v$ to be selected as a FN of $u$ is:

$$W_{uv} = R_{uv}(\sum_{w \in N_v} |M_{vw} - M_{uv}|).$$

$|M_{vw} - M_{uv}|$ is the number of elements left in $M_{vw}$ after removing the elements in $M_{uv}$ from $M_{vw}$.

Nodes that have more non-conflict channels and more links to forward the packets will have a larger weight.
Basic Greedy Algorithm

- **Candidates**: The forwarding nodes are selected from the downstream neighbors, in terms of ETX (estimated transmission count) metrics.

- The ETX of a single link is:

\[
\frac{1}{(1 - p_r) \times 1 - p_f}
\]

\(p_r\) and \(p_f\): the loss probabilities of the link in the forward and reverse directions.
**Basic Greedy Algorithm**

- Intuitively, the easiest way to select a forwarding set is to apply the greedy algorithm.
- To calculate the forwarding node (FN) set $F_u$ of $u$:
  - Select the node with the largest weight from the remaining candidates.
- **Advantages**: *Simple, less complicated, and relatively reliable.*
Greedy Algorithm With One Backtrack

- The basic greedy algorithm is very straightforward and easy to implement.
- However, it is likely that the size of the FN set selected by the greedy algorithm is too small.
- We provide one backtrack scheme for the greedy algorithm.
Greedy Algorithm With One Backtrack

- During every loop in which one node is selected into the FN set by the greedy algorithm, we keep a backtrack node for it.

- When necessary, the backtrack scheme will go back one step, replace one element, and rerun the greedy algorithm from that point.
Greedy Algorithm With One Backtrack

- The greedy algorithm for selecting $F_u$ for node $u$ with backtrack list $L_B$ maintained is:
  - Every time a node is selected with the maximum weight, maintain a backup node with the second maximum weight;
  - The backup node is later used for backtrack.
- Overlap is possible between the two lists, $F_u$ and $L_B$. 
Greedy Algorithm With One Backtrack

- When the $F_u$ cannot meet the requirement, e.g., the size requirement $\tau$, the backtrack algorithm is used.
  - Starting from the end of the FN list, replace with the corresponding backup node;
  - Repeat the greedy algorithm until the size constraint is satisfied.
**Greedy Algorithm With One Backtrack**

Some highlights are:

- We use the size requirement, which can be extended for other requirements.
- We only maintain one backtrack list here. Of course, more backtrack lists can be maintained, if one cannot find the appropriate FN set.
- The complexity of this backtrack scheme is low, since only one node is maintained for each node in the FN set.
Maximum Weighted Independent Set Algorithm

- The previous two algorithms cannot ensure that the selected FN set is the one with the maximum weight.
- Next part → the algorithm that gives the optimal result as of the overall weight.
Maximum Weighted Independent Set Algorithm

The process of the Maximum Weighted Independent Set Algorithm is:

- **Step 1**: Construct a graph among the candidate nodes.
- **Step 2**: Divide the graph into *modules*.
- **Step 3**: Recursively find the maximum weighted independent set.
Maximum Weighted Independent Set Algorithm

First, here are two definitions:

**Definition**

Given node $u$, its neighbor set $N_u$, and the distance threshold $\sigma$, we define a graph $G_u(\sigma)$, where

1. $v$ is a vertex in $G_u(\sigma)$, iff $v \in N_u$, $v$ is smaller of ETX to the destination than $u$ and satisfies ETX constraint;

2. an edge exists between two vertices, $v$ and $w$, in $G_u(\sigma)$ iff $d_{vw} > \sigma$,

**Definition**

Given a $G_u(\sigma)$, suppose $U$ is a subset of the vertex in $G_u(\sigma)$. For a node $v$, which is a vertex of $G_u(\sigma)$ and $x \notin U$, $x$ “distinguishes” $U$ if $x$ has both a neighbor and a non-neighbor in $U$. $U$ is a module if it is indistinguishable for the vertices outside $U$. 
Maximum Weighted Independent Set Algorithm

- An example for modules in a graph:

![Graph Image]

- The node $x$ distinguishes the node set $\{z, w\}$, since $x$ has both a neighbor and a non-neighbor in $\{z, w\}$. $x$ cannot distinguish the node set $\{y, v\}$ since neither $y$ nor $v$ is $x$'s neighbor.

- Two modules: $Q_1 = \{y, v\}$ and $Q_2 = \{z, w, x\}$. 
Performance Evaluation

- We compare the three algorithms by varying different network parameters:
Conclusion

• We consider the FN set selection problem in CRNs under the opportunistic routing.

• Three algorithms are proposed: the basic algorithm, the greedy algorithm with one backtrack scheme, and the maximum weighted independent set algorithm.

• The simulation results show desirable performance of our algorithms.
Thank you!