Implementation of Dynamic Spectrum Allocation for Cognitive Radio Networks based on Iterative Water Filling in OMNeT++/MiXiM

D’HONDT Sébastien
SCHEERS Bart - LENIR Vincent

OMNET++/MIXIM Workshop - 21 March 2011
Plan

• Introduction
• Presentation of IWFA
• Implementation of IWFA
• Simulation results
• Conclusion
Introduction
Introduction

→ Potential approach : Cognitive Radio (Dynamic Spectrum Allocation)
How to achieve Dynamic Spectrum Allocation?

→ Use of **Game theory**

> Game theory is a mathematical modeling of strategic situations (= game), in which an individual’s choice depends on the choices of the others. The choice is evaluated by an objective function »

→ Cooperative vs. **competitive** approach

> Nash equilibrium (NE) can be reached. In a NE, no player can increase the outcome of its objective function by unilaterally changing its strategy »

→ Iterative Water Filling Algorithm (IWFA)

> IWFA gives a sub-optimal allocation of the power by distributing the total power available at the transmit nodes on different sub-channels »
Introduction

Implementation of IWFA in OMNET++/MiXiM

→ IWFA has already been implemented in C/C++, matlab,…

…but never in an event-driven simulator

→ Improvement of IWFA thanks to the reduction of transmitted power

Total transmitted power is reduced (increased) if this power is higher (lower) than this necessary for reaching the desired data rate.

beneficial for :
  • economy of power
  • reducing interference with other networks
Presentation of the algorithm (1/3)

- **WFA**

<table>
<thead>
<tr>
<th>$N_c$</th>
<th>Amount of sub-channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_i$</td>
<td>Power delivered to sub-channel “i” by the transmitter</td>
</tr>
<tr>
<td>$T$</td>
<td>Target data rate for the network</td>
</tr>
<tr>
<td>$R$</td>
<td>Effective data rate in this network</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Channel property for channel “i”</td>
</tr>
<tr>
<td>$\sigma_i$</td>
<td>Standard deviation of the noise for channel “i”</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>Signal to noise ratio (SNR) gap</td>
</tr>
<tr>
<td>$\Delta f$</td>
<td>Sub-channel bandwidth</td>
</tr>
</tbody>
</table>

Minimize \( \sum_{i=1}^{N_c} P_i \)

subject to: \( T \leq R \)  \( \text{with} \)  \( R = \Delta f \cdot \sum_{i=1}^{N_c} \log_2 \left( 1 + \frac{P_i |H_i|^2}{\Gamma (\sigma_i)^2} \right) \)
Presentation of the algorithm (1/3)

- **WFA**

  Lagrange multiplier (dual form)

  \[ L(\lambda, T) = P_i - \lambda \left[ \Delta f \sum_{i=1}^{N_c} \log_2 \left( 1 + \frac{P_i \cdot |H_i|^2}{\Gamma \sigma_i^2} \right) - T \right] \]

  An additional variable (\( \lambda \)) is added in order to determine the maximum.

  First derivative

  \[ \frac{\partial L(\lambda, T)}{\partial P_i} = 1 - \lambda \left[ \Delta f \frac{|H_i|^2}{\Gamma \sigma_i^2 \cdot (1 + \frac{P_i \cdot |H_i|^2}{\Gamma \sigma_i^2})} \right] \]

  The first derivative is taken equal to zero.

  Optimum

  \[ P_i = \left[ \frac{1}{\lambda \Delta f} - \frac{\Gamma \sigma_i^2}{|H_i|^2} \right]^+ \]

  An optimum of the dual form is also an optimum of the initial form.
Presentation of the algorithm (1/3)

- WFA

\[
P_i = \left[ \frac{1}{\lambda \Delta f} - \frac{\Gamma \sigma_i^2}{|H_i|^2} \right]^+
\]

**ANALOGY:**

The water (= the power) has to be distributed in the reservoir (= the normalized noise in the sub-channels).

\[
\begin{align*}
\text{if} & \quad \frac{\Gamma \sigma_i^2}{|H_i|^2} \geq \frac{1}{\lambda \Delta f} \rightarrow P_i = 0 \\
\text{if} & \quad \frac{\Gamma \sigma_i^2}{|H_i|^2} < \frac{1}{\lambda \Delta f} \rightarrow P_i = \frac{1}{\lambda \Delta f} - \frac{\Gamma \sigma_i^2}{|H_i|^2}
\end{align*}
\]
Presentation of the algorithm (2/3)

- **IWFA**

  - Two couples of users
  - Communication through the same sub-channels
  - Noisy environment

Both couples want to ensure an optimal communication

---

**Network 1**

- Water Level
- Interferences
- Noise

**Network 2**

- Water Level
- Interferences
- Noise
Presentation of the algorithm (3/3)

- IWFA with Power Control

Effective rate ($R_j$) > target rate ($T_j$) $\Rightarrow$ Transmission power ($P_j$) ↓

Effective rate ($R_j$) < target rate ($T_j$) $\Rightarrow$ Transmission power ($P_j$) ↑

+ Battery saved
+ Decreases potential interferences
To implement the IWFA in Omnet++/Mixim, we need to:

**access** and **control multiple channels** simultaneously.

→ A first possible approach: use the signal class of MiXiM
→ Our approach: we extend existing modules in the connection manager
Extension of cGate into cMultiChannelGate

Before

\[ \text{Tx ConnectTo(Rx)} \]

Extension allows to specify the number of the sub-channel used for the transmission

Existing functions must be adapted with this supplementary parameter

After

\[ \text{Tx ConnectTo(Rx)} \text{ through subchannel } n \]
Customization of the connection manager

→ The connection manager doesn’t only manage the connection between nodes, but also the sub-channels that can be used for that purpose.

→ Addition of the sub-channel dimension to several functions (to specify the sub-channel through which a message has to be sent).

"SendToChannel()", "ConnectTo()", and "DisconnectFrom()"

→ Necessity of creating some new elements in order to stock data related to the execution of the algorithm.

3 new maps : "Node", "Waterfilling", and "Receiver"
Implementation of the simulation

Regulation of the power for each channel

**Map Node**

<table>
<thead>
<tr>
<th>MultiChannelNicEntry*</th>
<th>int</th>
<th>double</th>
<th>double</th>
<th>int</th>
<th>double</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>indexchannel</td>
<td>pathloss</td>
<td>power</td>
<td>network</td>
<td>status</td>
<td>staticrandom</td>
</tr>
</tbody>
</table>

**Map Waterfilling**

<table>
<thead>
<tr>
<th>double</th>
<th>double</th>
<th>int</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxpower</td>
<td>constraintpower</td>
<td>iteration</td>
<td>targetrate</td>
</tr>
</tbody>
</table>

**Map Receiver**

<table>
<thead>
<tr>
<th>int</th>
<th>double</th>
<th>double</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexchannel</td>
<td>noiseandinterference</td>
<td>powerowntx</td>
<td>pathlossowntx</td>
</tr>
</tbody>
</table>
Implementation of the simulation

Regulation of the power for each channel

Map Node

<table>
<thead>
<tr>
<th>MultiChannelNicEntry*</th>
<th>int</th>
<th>double</th>
<th>double</th>
<th>int</th>
<th>double</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td></td>
<td>indexchannel</td>
<td>pathloss</td>
<td>power</td>
<td>network</td>
<td>status</td>
</tr>
</tbody>
</table>

Map Waterfilling

<table>
<thead>
<tr>
<th>double</th>
<th>double</th>
<th>int</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxpower</td>
<td>constraintpower</td>
<td>iteration</td>
<td>targetrate</td>
</tr>
</tbody>
</table>

Map Receiver

<table>
<thead>
<tr>
<th>int</th>
<th>double</th>
<th>double</th>
<th>double</th>
</tr>
</thead>
<tbody>
<tr>
<td>indexchannel</td>
<td>noiseandinterference</td>
<td>powerowntx</td>
<td>pathlossowntx</td>
</tr>
</tbody>
</table>
Three main functions are defined for the execution of IWF:

- **CalcPower()**: fill in the map linked to the node class + updating pathloss value (function called with `UpdateNicConnection()`)

- **IterationInnerLoop()**: distributes the available power on the different sub-channels in order to maximize the data rate for this network.

- **IterationOuterLoop()**: compares Target and Real data rate. This function determines if the total available power must be decreased or increased.
Implementation of the simulation

Scheduling of those functions:

Calcpower() executed with UpdateNicConnection()
→ Every time a node has moved

Periodic scheduling of the inner and outer loop
Some functions of the connection manager have been customized.
→ Two events have been added to the existing ones:
  • EXECUTE-IL
  • EXECUTE-OL

Those events are declared in the "initialize()" function and defined in the "handleMessage()". Their execution is planned thanks to the "scheduleAt()" function.
The convergence of the inner loop is mostly reached after five iterations.
A realistic execution of the inner loop can be assumed every 0.5 second.
Simulation results

- 2 networks (1 static + 1 mobile: 90km/h)
- 2 nodes/network
- 4 sub-channels
- Max transmitted power: 10W
- Target data rate: 64kbps
Simulation results
Simulation results

- Convergence of the outer loop

Graph showing:
- Total Power delivered by the transmitter (W)
- Data Rate (kbps) vs Time (s)
Simulation results
Conclusion

Adaptation of cGate to make possible the use of multiple channels

Creation of maps for stocking data related to the algorithm

Implementation of functions and events necessary for the execution of IWF

Validation of the simulation in the case of 2 networks with 1 Rx and 1 Tx

This implementation will allow the study of the IWFA in more realistic environment

It can be used as a framework to compare different game theory based algorithms
Questions