SIMULATION ENVIRONMENT FOR DYNAMIC COGNITIVE RADIO NETWORK ARCHITECTURE: A SURVEY

Alan Yudhahutama¹, A.K.M Muzahidul Islam², and Nafees Mansor³

¹School of Electrical Engineering and Informatics (SEEI), Institut Teknologi Bandung (ITB), Bandung, Indonesia
²,³Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia (UTM), Kuala Lumpur, Malaysia
¹alan.yudha@hotmail.com, ²akmmislam@ic.utm.my, ³nafees@nafees.info

ABSTRACT

Due to the growing interest on Cognitive Radio Network (CRN), researchers have started spending much time on determining a suitable network simulator to verify their developed system architecture, protocols, etc. Sometimes, it requires even months to determine a simulator which could serve the best after trying a number of simulators. In this paper, we study and compare network simulators. Based on our primary study outcome, we further provide a deeper comparison between NS-2 and OMNeT++ simulators since their behaviors towards CRN are similar. It is observed that so far there is no best network simulator to serve CRN. However, OMNeT++ could be the best available network simulator to choose for CRN.

Keywords— Cognitive Radio Network, Dynamic Architecture, Simulation, Simulator, NS-2, NS-3, OMNetT++, TOSSIM, J-SIM.

1. INTRODUCTION

The research of Wireless Network is undergoing a paradigm shift from the traditional usage of computation network to a smart-and-adaptable radio network, mostly known as cognitive radio network (CRN) [14]. Combining a dynamic architecture [12] into a CRN is estimated as the new solution for cognitive implementation for mobile devices and disaster-ready network [14]. As the research goes through the final phase, the theorem needs to be verified using simulation. This kind of Network Simulation is usually done in a network simulator [14].

However, as the area of dynamic CRN research is still considerably new, there is lack of reference available for deciding which network simulator should be used. Most of the researchers end up spending a lot of time by trying each simulator before coming up to a conclusion.

In this paper, we discuss details of network simulator regarding cognitive-related protocol capabilities, especially for a dynamic network. The contribution of this paper is as follows:

- It investigates the general specification of CR node and dynamic CRN simulation
- It investigates the CR-related ability and comparison for each simulator
- It proposes network simulators for dynamic CRN
- It develops a framework for dynamic CRN in OMNeT++/MiXiM

The paper is organized as follows. Section II describes the general dynamic CRN simulation specification which is mostly used for CRN architecture [8], [13]. Section III describes different kinds of network simulator and shows comparisons between them. Section IV describes and compares framework that can be developed through NS-2 and OMNeT++. The development of CR Framework is further discussed in Section V. Finally, the conclusion is drawn in Section VI.

2. GENERAL DYNAMIC CRN SPECIFICATION

To determine which network simulator should be used, a classification of several simulation characteristic in dynamic CRN should be made in order to be traced and compared. Classification begins by detailing the desired parameter and simulation specification of this kind of network, which are the dynamic network and the CR node.

The specification of the dynamic network simulation is described as follows:

- Flexibility in implementing custom routing protocol
- Ability to simulate more than thousand nodes effectively
- Supported by GUI, for debugging method
- Ability to determine efficiency, in terms of time, power, message transfer, and delivery result

The specification of the CR node simulation is describe as follows:

- Flexibility in implementing custom PHY and MAC algorithm
- Ability to implement custom sensing algorithm
- Support default communication protocol
- Supported Cognitive
3. NETWORK SIMULATOR COMPARISON

Network simulator which will be used for comparison here is the one which popular among research of telecommunication networks. Comparison was made with the aim to provide a brief description and pursed the network simulator which will be discussed later. Table below shown the result based on the comparative study of literature:

Table 1. General Comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognitive Framework</td>
<td>Yes, CRCN</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Can be developed</td>
</tr>
<tr>
<td>IEEE 802.22 Module</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Parameter</td>
<td>Radio Energy</td>
<td>No, focus on radio energy</td>
<td>Yes, with Power TOSSIM</td>
<td>Radio Channel and Power Consumptions</td>
<td>Radio Channel and Power Consumption</td>
</tr>
<tr>
<td>General/Specific</td>
<td>General</td>
<td>General</td>
<td>Specific</td>
<td>General</td>
<td>Specific</td>
</tr>
<tr>
<td>Simulator/Emulator</td>
<td>Simulator</td>
<td>Simulator</td>
<td>Simulator</td>
<td>Simulator</td>
<td>Simulator</td>
</tr>
<tr>
<td>Open Source</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes, for education-purpose only</td>
</tr>
<tr>
<td>Languages</td>
<td>C++</td>
<td>C++ / Python</td>
<td>NesC</td>
<td>Mathematical Modelling Language</td>
<td>NED &amp; C++</td>
</tr>
<tr>
<td>Platform</td>
<td>Linux, Unix</td>
<td>Linux, Mac OS, Unix</td>
<td>Linux, Unix</td>
<td>Java</td>
<td>Linuc, Mac OS X, Windows</td>
</tr>
<tr>
<td>GUI</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Low</td>
<td>-</td>
<td>High</td>
<td>-</td>
<td>High</td>
</tr>
<tr>
<td>Ability to implement new protocol</td>
<td>Yes, but difficult</td>
<td>Yes, but difficult</td>
<td>Application Only</td>
<td>Yes, but difficult</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Based on the table shown above (Table 1), it is clearly described that OMNeT++ and NS-2 are custom best-suited for CR development. Both of them have a CR-related framework that can be developed for later use. However, the above comparison is general and not related to point of efficiency in large network simulations. Thus, a deeper comparisons need to be made in term of efficiency. It needs to know how efficient both of simulator tools when simulating large number of nodes. Efficiency in this case means time taken when simulating a network cases.

To get information about this point, a dynamic topology and scenario are tested in both simulator tools. The dynamic architecture is based on the development of paper [12] and has the rules shown below:

1. Consider that there are randomly spread nodes in a given area. The number of the nodes is an input from the user. Nodes are then connected with each other if they are in each other’s range.
2. There are four types of nodes:
   a. DCBS (Dynamic Cognitive Base Station)
   b. CH (Cluster Head)
   c. GW (Gateway)
   d. CM (Cluster Member)
3. Every nodes that has connection with the DCBS becomes CH
4. The rest of the rules are similar to dynamic clustering shown in the Section IV [12].

The simulation is done in OMNeT++ and NS-2. It gives following result:

Table 2. Simulation Result

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Sim Time OMNeT++ (s)</th>
<th>Sim Time NS-2 (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.257</td>
<td>0.951</td>
</tr>
<tr>
<td>200</td>
<td>1.031</td>
<td>2.824</td>
</tr>
<tr>
<td>300</td>
<td>2.957</td>
<td>5.025</td>
</tr>
<tr>
<td>400</td>
<td>6.437</td>
<td>7.796</td>
</tr>
<tr>
<td>500</td>
<td>11.73</td>
<td>13.588</td>
</tr>
</tbody>
</table>

Fig.1. Dynamic Topology of Simulation

Fig.2. Graph Showing the Simulation Result
4. NS-2 CRCN AND OMNET++/MIXIM

Both simulators need additional frameworks to do CRN related simulation. Unfortunately, both of them have several concerns as shown below:

a. NS-2 with CRCN (Cognitive Radio Cognitive Network) [5]
   1. MAC layer is not aware of multiple channels created for each radio. Channel selection must be made in routing layer instead of MAC layer, which is conflict with most current MAC protocol designs. Thus, additional work is required in MAC to support multiple channels in MAC, which can be found in next section.
   2. MAC layer address confliction is introduced by the related works.
   3. Since channels are created in the same way, these radios and channels have the same radio and spectrum characteristic, which is not sufficient to support the simulation in heterogeneous CR network.
   4. The radio and channel number are equal. Some routing and MAC algorithms simulations such as common control channel based algorithms are not supported if we only adopt the previous contribution work alone.
   5. CRCN comes with a quite fixed packaging and only compatible with old version of NS-2, means it needs quite development for implementing custom algorithm.

b. Omnet++ with MiXim, INET, or MATLAB
   1. Doesn’t specially made for CR simulation, thus need to create own CR framework based on the desired node architecture.
   2. Best suited for implementing cognitive protocol in lower layer. [6]

Both simulator tools have ability to accomplish objective statement. The table below shows more comparison of other features and properties for both simulator tools outside the objective statement. This table made for giving a deeper insight in doing development on the simulator.

<table>
<thead>
<tr>
<th>Table 3. Properties of Each Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
</tr>
<tr>
<td>GUI</td>
</tr>
<tr>
<td>IDE</td>
</tr>
<tr>
<td>License</td>
</tr>
</tbody>
</table>

5. CR FRAMEWORK IN MIXIM

As the study of desired cognitive radio network developed, a new challenge is newly recognized. Here are several challenges in the desired cognitive radio network:

1) The desired standard network for cognitive radio is IEEE 802.22. However, OMNeT++/MiXiM don’t support that module. As the searching for the network simulator that support that module going, an unfortunate fact is revealed: no network simulator currently supports that module.

2) The desired cognitive simulation is very customizable while no network simulator provides a very customizable framework for cognitive radio development. NS-2 provide CRCN frame for cognitive radio, but it is used for older version for NS-2 and its latest update is at 2009—which means several protocol are still need to be developed. Also, there are many several limitations in CRCN which has been reviewed in the chapter IV. This means a new framework must be made. As a new framework is choosen as current best solution, 802.11 is used for the main protocol development. It means that the framework is not built from scratch, but from the development of 802.11. IEEE 802.11 is choosen because of three reasons:
   a) OMNeT++/MiXiM already support that module.
   b) It is based on wireless protocol, which is closest to the desired CR network.
   c) IEEE 802.11k support channel sensing template.

The CR Framework in the OMNeT++/MiXiM is then made based on reference [10]. Figure 3 shows the architecture of CR Node which will be implemented in OMNeT++/MiXiM.
Here is what it looks like in the OMNeT++ implementation:

**c. Flexibility in simulating heterogeneous cognitive network.**

NS-2 CRCN comes with a fixed packaging while OMNeT++ comes with flexible cognitive development. Hence, since the CRN protocol algorithms are not developed yet, it will be safer to use OMNeT++ as the simulator.

**ACKNOWLEDGEMENT**

This work is partially supported by MJJIIT Research Grant, with Vote No. 4J044, Ministry of Higher Education (MoHE) of Technology (MJIIIT) of Universiti Teknologi Malaysia (UTM) Year 2012-2013.

**REFERENCES**