FORMAL VERIFICATION OF A DISTRIBUTED ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS WIRE ON A LARGE SCALE

January 12, 2015

Supervisors:
- Saadi BOUDJIT
- Laure PETRUCCI

Author: NGUYEN Viet Hai
Contents

1 Abstract 1

2 Introduction 1
  2.1 Wireless sensor network . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
  2.2 Formal modelling and Verification . . . . . . . . . . . . . . . . . . . . . 3

3 Presentation of Energy Efficient Routing in Wireless Sensor Networks 3
  3.1 Wireless Sensors Networks and application . . . . . . . . . . . . . . . . . . 3
  3.2 Constraints of WSNs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  3.3 Structure of the network . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
    3.3.1 Network technique update . . . . . . . . . . . . . . . . . . . . . . . 4
    3.3.2 Motivation and assumptions . . . . . . . . . . . . . . . . . . . . . . 6
    3.3.3 SN-MPR Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
  3.4 Perforance evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

4 Petri Nets 11
  4.1 Petri net and subclasses . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
    4.1.1 Syntax . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
    4.1.2 Firing Rule . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
    4.1.3 Modelling Petri net . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
    4.1.4 Properties of Petri nets . . . . . . . . . . . . . . . . . . . . . . . . 15
  4.2 Coloured Petri net . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
    4.2.1 Definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
  4.3 Symmetric net with bags . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
  4.4 Cosyverif tools . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

5 Modelling the protocols 17
  5.1 Choosing main properties and assumptions . . . . . . . . . . . . . . . . . . 18
  5.2 Choosing Symmetric net with bags . . . . . . . . . . . . . . . . . . . . . . 19
  5.3 Applying into Petri Net . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
    5.3.1 Properties of the modelling . . . . . . . . . . . . . . . . . . . . . . 19
    5.3.2 Graph Modelling . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

6 Conclusion and future works 21
  6.1 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
  6.2 Future work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
1 Abstract

Nodes constituting a wireless sensor network (WSNs) are capable of communicating in ad hoc mode using a routing protocol and send information on the environment (temperature, fire detection, measurements in an agricultural field...). However, the use of routing the original proposed protocols for the ad hoc wireless network is not desirable in a wireless sensor network where energy consumption of the nodes is a key factor in the network lifetime. Indeed, many functions in routing protocols for ad hoc wireless networks are linked to the mobility management nodes in a wireless sensor network are often static. In this context, the network L2TI team focused on the data dissemination problem with energy saving in Wireless sensor networks on a large scale with Sink mobile. A distributed routing protocol called SN-MPR was proposed. This protocol is used to limit the spread of messages control over the location of Sink only to areas affected by the mobility of the latter. Another variant of the protocol, called SN-MPR duty-cycle, saving energy sensors allowing those that are not MPR turn off their radios when they have no data to be transmitted to the Sink. Simulation results showed that the protocol allows energy consumption perfectly distributed in a network Wireless sensors with movable Sink and thus increases the lifetime of the network. This course fits around the routing protocol for wireless sensor networks previously mentioned.

The objective of this report is to complete the assessment of performance obtained by simulation, by formal verification of the protocol:
- First, I become familiar with SN-MPR protocol and routing in wireless sensor networks.
- In a second time and with the help of the team LCR laboratory LIPN the trainee will be assigned the task of verifying the proper operation of the Protocol and to quantify robustness of formal verification methods for distributed protocols.
- Verification is to prove the suitability of the routing. The quantization is to measure the Sink maximum speed of the protocol for which still meets its specification. The formalism of Petri nets and their subclass, abstract will be used.

2 Introduction

In this work, there are two main relate work need to be done for complete the goal. The SN-MPR protocols is the main object that need to be verified on my work. This must be deeply understand the properties of SN-MPR networks, how it work. By deeply understanding it, that will be more advantages for analyst into formal verifying tools. Petri net and its subclass is the tools for proving SN-MPR protocol and routing in wireless sensor networks. Working on Petri net and using there subclass is must for my work’s goal. In additional, Cosyverif tools [2] is advised to using on my work.

2.1 Wireless sensor network

- Architecture
  
  Wireless sensor network is a succeed of development in micro-electronics technology,
that created inexpensive wireless sensor device carrying wide of range. We can implement application of wireless sensor network in wide field like agriculture monitoring, habitat monitoring, health controlling, battle field, etc. In a Wireless sensor network, we have many devices for collecting data which are called wireless sensor device. The wireless sensor device includes a micro-controller, wireless transceiver, memory, battery and sensors which help the network for environmental data collection. The sensors of these device are supposed to sense pressure, temperature, humidity, etc. After collecting environmental data, the device will transmit data through wireless network to data center for processing data or transmit to another data center.

- Characteristics

Wireless sensor network can help human for updating data as soon as possible without moving to direct place for data collection, people do not need to reach danger place or to go to place permanently, all tasks will be done by placing wireless sensor for collecting data. But there are some constraints of WSNs that the networks will need to be developed more for more effective. The problems of WSNs consist of reliability, data delivery delays, data aggression, efficient data dissemination to the sink, and energy constraint. In this work, we mainly focus on Energy Efficient Routing in Wireless Sensor Networks. Why is Energy Efficient?

In the wireless sensors networks, we use small device which is small battery, so the energy will drain out rapidly when we use permanently. Cause it drain out very quickly and can not replacing soon, all the networks will die very soon and networks will not run effectively. So the work for Energy Efficient is very important to make sure that networks will work perfectly. The work will run on small scale to large scale deployments for transfer monitored information sink, ensure connectivity, and increase network life time. Then we can propose energy efficient data dissemination mechanisms.

- Attributes and Protocols

For proposing energy efficient data dissemination mechanisms, our work mainly work on data dissemination in large scale wireless sensors networks. That we consider the data center that collect all data of the network is the SINK of the networks. All the others is the node of the networks, which is collect data from environment. All node of network is a static node. With the Sink of the networks, a mobile sink will be chosen. Although, the static sink have advantages of low control traffic overhead, but it will have problem with energy hot spot problem. The problems will make network disconnect rapidly, it does not consists of our goal for energy consumption. Therefore, the mobile sink will be the chosen solution for this project. That may be can increase control overhead but will have better energy distribution. Our work will verify that mobile sink will not cost more than static sink too much. In updating techniques for network, we will use Pure Flooding, Backbones structures and Tree-based data dissemination. That using Sinks as a root and located Sinks for updating data, and try to minimize the network update
2.2 Formal modelling and Verification

In this work, for modelling and verification on WSNs will be done on Petri nets. By advice of professor Laure Petrucci, the work will work mainly on symmetric net with bags. The Symmetric net with bags is an extension of symmetric net which is a subclass of coloured petri net. Petri net is a mathematical modelling languages, which describes distributed systems. Petri net using a directed bipartite graph, which have transitions, places. Transition described by bar represents the event that occur when running machine, network, etc. Places is a circle which is condition of when an events are fired. Arcs connect place and transition, which is described by an directed arrows. An arc will never connect between two places or two transition. In places, there are many tokens, that represent a configuration of the net called a marking. In an abstract sense relating to a Petri net diagram, a transition of a Petri net may fire if it is enabled, i.e. there are sufficient tokens in all of its input places; when the transition fires, it consumes the required input tokens, and creates tokens in its output places. A firing is atomic, i.e., a single non-interruptible step. By providing the power of modelling and analyzability, petri net can give the solution of many system with solving three problem in an network or automatic machine.

For modelling and verification of the wireless sensor networks, Petri net is not enough, because there are many variable and event that need to be considered inside the network. To solving the problems, symmetric net with bag is choose for this. In symmetric nets with bags, a bags is a multi sets of conditions or tokens.

3 Presentation of Energy Efficient Routing in Wireless Sensor Networks

3.1 Wireless Sensors Networks and application

Wireless sensor network is a spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. This is built by many node for transferring data to the data center. By using many node for building the network, people can get data through sensor of the node permanently and instantly in many place without travelling. They just need control the data center for these information. The sensor wireless network has been used in many fields like health care monitoring, process management, environment sensing, forest fire detection, battle fields, etc. But there are still many constraints of wireless sensor network.

3.2 Constraints of WSNs

- Reliability
• Data delivery delays
• Data aggregation
• Efficient data dissemination to the sink
• Energy constraint

This is the main constraint that my work will focus in. In the wireless sensors networks, there are just a small device with limited battery and difficult to replace battery immediately. Because we use small device for the sensors and setting these device in a large range or risky place. Therefore, we do not have many energy for each device and hard for replacing usually. In this work, we propose energy efficient data dissemination mechanisms for solving this problem. With this propose, we will consider about transfer monitored information to sink, ensure connectivity, and increase network life time.

3.3 Structure of the network
3.3.1 Network technique update

• **Backbone-base structure:** In such structures, nodes organize themselves into either clusters or virtual grids to create virtual a topology on top of the physical topology. In cluster-based approach, nodes organize themselves into clusters where mostly only one Cluster Head (CH) per cluster is responsible for data aggregation and query forwarding. An alternate approach is to organize the network into virtual grids on the basis of their geographic location in which one node per cell may perform data aggregation and communication with the network. [5][7]

![Figure 1: Backbone structure](image)

• **Tree-based dissemination structure:** There are two kind of tree-based dissemination structure: source-rooted tree and sink-rooted tree. In source-rooted tree, every nodes will detect node broadcasts events information to the network and creates a tree-rooted at itself. After that, the sink will use tree of source to retrieve information of each source. In the other way, sink-rooted will broadcasts source location to the network which creates a tree-rooted at the sink or sink assigned root node, and the reserve tree path will be use to delivery data to the sink. In these
two technique, there are pros and cons of each technique. In the source-rooted tree, there is a routing structure independent of the sink mobility as sink may use any tree branch to demand data from the source. But, the technique need large number of sources for making infeasible for applications. In contrast, sink-rooted tree have lower routing structure creation overhead. However, they will increase the control traffic overhead and networks energy consumption for periodic location updates to the network. In this work, these two technique will compose to each other and Our proposed MPR broadcast based algorithm maintains a dynamic data dissemination tree with minimum control overhead.

![Source rooted tree](source.png) ![Sink rooted tree](sink.png)

Figure 2: Sink-based and soured-based

- **Multi-Point Relay Broadcast**: This mechanism was proposed by Qayyum for mobile ad hoc networks[7]. The MPR broadcast mechanism reduces flooding costs by minimizing the number of message forwarding nodes. Exchange of hellos enables every node to obtain information about its one hop and two hop neighbor, referred to as $n_i1$ and $n_i2$ respectively. One should note that apart from hello messages, neighboring nodes will only communicate if they have established a symmetric link between them.

Once, $n_i$ has discovered its one-hop and two-hop neighbors, it chooses a minimum subset among its $n_i1$ neighbors, such that node $n_i$ can access all its two hop neighbors i.e. $n_i2$ via this subset. These minimum number of one hop neighbors are referred to as MPRs of node $n_i$ message generated by $n_i$ will only be forwarded by its MPRs, whereas its remaining non-MPR $n_i1$ neighbors do not forward them. Thus, the MPR broadcast algorithm minimizes network update costs by reducing the number of retransmissions of broadcast packets. One should note that a node can avoid itself from being selected as a MPR by its neighbors by lowering its willingness field value. It can also refuse to act as MPR by setting its willingness to 0. This field can be used by the nodes which have less battery power to avoid being selected as a MPR. [5]
3.3.2 Motivation and assumptions

For building the proposed network model, network follows these assumption:

- Once deployed nodes remain static
- Nodes are not aware of their geographical position
- Network topology is relatively dense
- Node transmit with same transmission power
- Node transmit monitored data to a single mobile sink

3.3.3 SN-MPR Algorithm

The algorithm is modified based on MPR algorithm[7] for resource constrained WSNs. This section discusses and explains the modified for proposing SN-MPR Algorithm[5].

1. Neighbor discovery: Based on neighbor discovery of MPR Algorithm, hello massages will be send for this function. But there are a little different in SN-MPR, that the hello messages will only be sented at the initialize of the network. After created the MPR network, there are no need for broadcast the hello messages. There are two type of neighbor discovery: sensor neighbor discovery and sink neighbor discovery. In sensor neighbor discovery, it just executed at the network initialization phase as sensors remain static after deployment. Newly deployed nodes start exchanging hello packets with 2 seconds interval[5]. After all node discovered their 1 hop and 2 hop neighbor, the function of sending hello messages will be shutdown. In the sink neighbor network, sink will work nearly like another normal sensor, but there are no terminal of sending hello messages of the sink. Since sink always moves around the networks for collect data, it need to transmits hello messages every 2 seconds. By doing this, sink will notify all network the position that it located for receiving information from every sensor in the network. The location update technique of sink will be described at Sink location update.
2. **Sink location update:** In this step, the sink will broadcast a location update messages called *Sink location update messages (SLU messages).* A SLU message contains the ID of the sink, and if required, the sink may also append its queries in this message. When sink broadcasts a SLU packet, it is only retransmitted by only its MPRs, and then onwards by the MPRs of its neighbors and so on till the message update is transmitted to the whole network.[5]

![Figure 4: Sink location update processing](image-url)
Figure 5:

3. **Reverse Tree Route Configuration**: In [5], paper propose a reverse tree based data dissemination mechanism in order to enable the network to communicate with the sink. This mechanism is based on the SLU messages, every neighbor of the sink chooses SLU messages reverse propagation path to configure its next hop route towards the sink. Every next hop of the sink will receive its SLU messages, but only node that chose as MPR node will be the transmitters that will transmit the SLU messages through all the network topology. By using this mechanism, one node can receive more than one SLU messages for finding path to the sink from many different neighbors. However, in order to construct the reverse shortest-path sink-rooted tree, SLU receiving node should select as its next hop towards the sink the neighbor from which the SLU is received with minimum hop count, else the reverse tree will not be a shortest path tree and the resulting topology may have routing loops. [5]

4. **MPR role exchange**: After choosing MPR for transferring data network, there are one problems that MPR node will consume more energy than the other one.
These nodes consume more energy, because it need to transfer its own data and the normal node’s data to the sink. To fixed this situation, when a node $n_i$’s falls below a certain threshold, it will send this information in the hello messages to neighbors. Neighbors upon receiving, and locally exchange few hellos. This helps in balancing the power consumption in the network as nodes this hello observe the lowered willingness of $n_i$ reselect MPRs with higher residual energies.

5. **Primitive buffer**: When the sink has changed position for collecting data, there will be loss data from the sensor node, because the missing of the sink at the old position. For repairing this situation, in [5] propose a mechanism called primitive buffer for preventing loss data in changing sink’s position time. As an act of precaution, sinks one hop neighbors consider that sink may no longer be in their vicinity, and thus, they stop sending data reports to the sink. The one hope nodes of sink will start buffering data that they have not send to the sink yet, and they are also buffering the incoming data too. After they can detect the sink new position, they will send all their data to the sink. Figure-6b describes how sinks neighbors start acting as temporary root nodes, when they do not receive sinks hellos. Figure-6c shows temporary root nodes returned to the normal state, after they have reconfigured their route, and transmitted all buffered packets to the sink.

![Figure 6: Primitive buffer mechanism](image)

### 3.4 Performance evaluation

- Implemented in Network Simulator.
Figure 7: Implemented in Network Simulator.

- Control traffic overhead of SN-MPR.

![Graph showing control traffic overhead of SN-MPR.]

Figure 8: Control traffic overhead of SN-MPR.

- Data Delivery Delay vs Sink Speed.

![Graph showing data delivery delay vs sink speed.]

Figure 9: Data Delivery Delay vs Sink Speed.

- Energy Distribution Efficiency of SN-MPR.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>1000m x 1000m</td>
</tr>
<tr>
<td>Randomly deployed nodes</td>
<td>400</td>
</tr>
<tr>
<td>Transmission range</td>
<td>80m</td>
</tr>
<tr>
<td>MAC</td>
<td>802.11</td>
</tr>
<tr>
<td>Sink speed</td>
<td>2-20m/s</td>
</tr>
<tr>
<td>Sources</td>
<td>64B data packets every 10 sec.</td>
</tr>
<tr>
<td>Sink Mobility model</td>
<td>random</td>
</tr>
<tr>
<td>Simulation duration</td>
<td>1000s</td>
</tr>
</tbody>
</table>
4 Petri Nets

4.1 Petri net and subclasses

Petri net can also called Place/Transitions Network. It helps to solve problem with concurrency and asynchronous processing. With Petri net, we can prevents from race conditions, resource starvation and deadlocks which is caused by concurrency. It means petri net can helps to fix when the result of system behaviour is not good as expected caused by race conditions, good resourced management for running to prevent from resource starvation, and no more deadlocks to make machine stopping out of expected. For building a petri net, we use a graph describing it as $G := (V, E)$. With $V$ is a set of vertices and $E$ is a set of directed edges (arrows). In the V, there are two types of nodes. First is places describing by an eclipse, for simulating the states of systems or machines. Second is transition describing by a rectangle for simulating event in systems or machines. These edges of Petri net called arc. Finally, for running the Petri net, we need tokens inside places, which denoted y a solid dot. Tokens mainly describe resources in each systems. In each places that have number of tokens that we call marking, and if the marking is at the beginning if analysis that can be called initial marking. Moreover, there are properties of Petri net that is basis for their formal analysis as reachability, liveness, boundedness, etc.

Although, Petri net have many advantages in modelling systems and machines. But it is still very simple, and it can be simulate model similar process using, tokens are identical and tokens are too simple to describe additional data. In order to upgrades the power of petri nets, there are many net that is proposed as: coloured Petri net, time Petri net, symmetric net, etc. In my work, I work mainly on coloured petri net and symmetric net.
with bags. These two subclasses of petri net is more advantages in using data on tokens and using tokens lists in Places.

4.1.1 Syntax

Definition 1 (Petri Net [6]).
A Petri net is a 5-tuple \((S, T, F, W, M_0)\) such that:

- \(P = \{p_1, p_2, ..., p_n\}\) is a finite set of places.
- \(T = \{t_1, t_2, ..., t_m\}\) is a finite set of transitions.
- \(P \cap T = \emptyset \quad P \cup T = \emptyset\)
- \(F \subseteq (P \times T) \cup (T \times P)\) is a set of arcs.
- \(W : F \rightarrow \mathbb{N}\) is a weight function.
- \(M_0 : P \rightarrow \mathbb{N}\) is the initial marking.

![Figure 11: Example of Petri Net](image)

4.1.2 Firing Rule

For simulating states, behavior of the system, Petri net follows these rules for changing and that is called Firing rule. [6]

1. Transition is enable, and it can fire. And it is fire only if the tokens of input places reach the conditions of transitions.

2. When transition is fired, tokens inserted from input places and send it to output places. When remove and inserted, the number of tokens can be different.

Petri net can have a transition that does not need an input place, and that is called source transition and there is a transition do not have output place calling sink. Note that source transition do not consume any produce.
4.1.3 Modelling Petri net

This section will show how to modelling a system into Petri net and some method. Firstly for modelling a system, you must understand project system and properties. This is the following step for analysis system:[4]

- Understand the structure.
- Choose key elements.
- Find interaction.
- Secondary elements.
- Devide into module.

In [1], modelling systems into petri net can be used by some typical situation. here are some method examples of modelling in to petri net.

- **CONCURRENCE:** In this typical situation, there are two petri net graphs that firing simultaneously, and there are not modifying the state of each others. These two graph are run concurrently.

![Figure 12: Example of concurrency](image)

- **SYNCHRONIZATION:** In synchronization, the transition have conditions that need to be get from two or more for enable. In the below figure that $t_3$ whose firing requires a token both in $p_2$ and $p_4$. 

13
• **LIMITED RESOURCES:** For handling limited resource, and reusing it, in modelling, there are an adding place for buffering resource. By using this, we can avoid the locking dead of exhausted resource in running system. Tokens in each place is use as resources.
• **SEQUENTIALITY:** A producer produces objects that are put into a buffer from which can be removed and consumed by a consumer. The consuming process must be in sequence with respect to the production process.

![Diagram of sequentiality](image)

**Figure 15:** Example of sequentiality

4.1.4 Properties of Petri nets

• **LIVENESS:** Transition that can be fired in any marking is a live transition. In the other, dead transition do not potentially firable in any marking.

• **SAFENESS:** A place is safe if there are no exceeding 1 in any of marking. If each places of PN is safe, PN is safe.

• **BOUNDEDNESS:** is a generalization of safeness. A place is boundedness when it do not exceed more than k in any marking, and PN is boundedness when every own places is k-bounded.

4.2 Coloured Petri net

Coloured petri net (CPN) is an extension of Petri net with more effective properties. The most useful properties of CPN is that the different of tokens between Petri net. In CPN, token that called coloured tokens, which means it can contains data value. With the value that tokens have, these can be investigated and modified by the occurring transitions. CPN is a combination of the strength of Petri nets with the strength of programming languages. Petri nets provide the primitives for the description of the synchronisation of concurrent processes, while programming languages provide the primitives for the definition of data types and the manipulation of data values.
4.2.1 Definition

In CPN there are tuple \( N = (P, T, F, \Sigma, C, W, E, G, I) \). You can reference syntax that contain in 4.1.1

- \( \Sigma \) is a set of color sets defined within CPN model. This set contains all possible colors, operations and functions used within CPN.
- \( C \) is a color function. It maps places in \( P \) into colors in \( \Sigma \).
- \( E \) is an arc expression that each arc \( f \subset F \) map the expression \( e \).
- \( G \) is guard function. This map the transition \( t \in T \) into expression \( g \). The result of this is boolean, means true or false.
- \( I \) is an initialization function. The initialization expression must evaluate to multi-set of tokens with a color corresponding to the color of the place \( C(p) \).

**Figure 16: Philosophers - example of CPN**

4.3 Symmetric net with bags

Symmetric net with bags (SNB) is a combined of coloured petri net with symmetric Petri net which is known as Well-Formed Petri Nets. SNB can have structured specification of complex systems like coloured petri net and using construction of a quotient state graph representation in symmetric petri net. The figure 17 is an example of SNB:

**Figure 17: Sale store - example of SNB**
The most advantage in the example of SNB is using bag for place domain. There are two bags is \textit{BagPeople} and \textit{BagGift}, these properties are built on the basic color classes of coloured Petri net. It is obvious that with bags manipulation, SNB provide a much more compact and natural way to model system than SN. More over, SBN allow production of a more compact quotient reachability graph in number of edges. So, using bags is more effective than express the symmetries of possible bindings of variables to values. In \cite{2}, you can find out more syntax of SBN for using in cosyverif tools, which will be presented in the next section.

4.4 Cosyverif tools

Cosyverif is a software environment whose goal is the formal specification and verification of dynamic systems. In this work, for modelling the system, cosyverif is a good choice for that. This is designed for: \cite{2}

- support different formalisms with the ability to easily create new ones,
- provide a graphical interface for every formalism,
- include verification tools called via the interface as a web service,
- offer the possibility for a developer to integrate his/her own tool, also allowing it to interact with the other tools.

This environment consists of two software tools: Coloane, the graphical interface, and Alligator, an integration framework based web services. It is enlarged with the existing verification tools developed in our laboratories (founding members or partners).

5 Modelling the protocols

In my modelling work, I mainly focus in the initial state of the Wireless sensor network, and the modelling graph will work on this state. This state is the most important state for setting up the networks when it is started. The modelling is mostly designed for SN-MPR algorithm running at the initial state when the network is stared. And the SN-MPR will have:

- \textit{Every initial node broadcast hello messages}
- \textit{Finding hope 1 and hope 2 for every node through hello message in the network}
- \textit{Choosing MPR-Node}
- \textit{Broadcast MPR-messages}
5.1 Choosing main properties and assumptions

Main Properties:

- Wireless sensor of course is the main properties, in this modelling we know them as nodes. There are three kind of node for simulate in the WSNs: Normal node, MPR Node and Sink node.

- Normal node: Node that just collect data from environment and send it to the data collector Sink Node. These node are permanently static. In these node, they have a list of there 1 hop and 2 hop node, and the MPR-node which is help them choosing the way for sending back data.

- MPR-node: These node is more likely as the normal node, but the main different between them is that they are sending not only their own data but also normal nodes’s data which are received the MPR-message from them. These node join the network and build like a main branch of the tree transferring data to the Sink node.

- Sink-node: This node is the data collector of the network, and especially this node is not static. This node is not collect data from environments, it just a data collector which receive from every other node in the network. And these node move around the network.

- List of the nodes: In these types of nodes, we determine that they always have an list for determining their one and two hope node, and the MPR list for sending MPR messages.

- After finding the MPR-node for each node of network, node will not send hello messages except Sink will send it every 2 seconds for announcing its own position. Every 5 second, Sink will broadcast the MPR-message for detect the root again

- In the initial of network mainly focus on SN-MPR algorithm. The functions of wireless sensor in WSNs that the modelling need to simulate are sending hello message, finding 1 and 2 hop nodes, choosing the MPR functions.

Assumption

- The network already know the position of every node except the sink. Although, in [5] we know that when initially every node will do not know their position of each other. But they are all statically positions, so I assume that they already have a plan of each other.

- The Sink will join the network after every node have setting up the network.

- The Sink will move randomly.
5.2 Choosing Symmetric net with bags

At the first time, I choose coloured petri net for modelling on this project, but coloured petri net have some disadvantages that can not solve our problems. Mostly on the list of the nets. Because using list in every node is hard to define in coloured petri net, but in symmetric net with bags that is very simple and more compact modelling.

5.3 Applying into Petri Net

5.3.1 Properties of the modelling

This section is describe the function of class, domain, variable in the modelling. For more understanding the syntax and structure of Symmetric, review in [2].

Class

- Node: This is the class for describe normal node and MPR node of the network. These node is identify by the number of the node. Syntax: Node is 1..10;

Domain

- ListNode: Using the bag symbolic for describe the list of 1 hop or 2 hop nodes. Syntax: ListNode is Bag(Node)
- Node1Hop: 1 hop node describe node that it belong to the list 1 hop of node. Syntax: Node1Hop is < Node, ListNode >. Node is describe the main node, and listnode describe the 1 hop list.
- Node2Hop: 2 hop node describe node that it belong to the list 2 hop of node. Syntax: Node2Hop is < Node, ListNode, ListNode > Node is describe the main node, and first listnode describe the 1 hop list, second listnode describe the 2 hop list.
- HelloMessage: Syntax: HelloMessage is < Node1Hop, Node >
- Syntax: Connode is < Node, Node >;

Variable

- Syntax: x,y,z in Node
- Syntax: h1,h2,h3,h4 in ListNode

Place (State) of the WSNs:

- Initial State: This is begin state, with Node is domain of this state and it contain all the node of network. Initial marking of this state is ¡Node.all¿.
- Connected: with assumption of knowing all the network before working. Using Connode in Domain like a link between these two hop. This state need to be marking at the initial.
• **Actived**: This state is represent all node of the network with using domain Node2Hop with 2 list of hop 1 and hop 2 connecting to the node. This state is marked at the initial with all node of the network and 2 list of connecting hop, which is null-able

• **1HopIdentify**: 

• **ActiveNodeComplete**: 

**Transition and Guards:**

• **SendingHLL**: No guard for this transition. Transition will receive input from Initial state, connected, Actived Place and send output to Actived, 1HopIdentify and Connected place

• **ReceiveHLL**: No guard for this transition. Transition will receive input from connected, Actived, 1HopIdentify Place and send output to the same place.

• **FinishInit**: Guard for this transition: \[ h_3 \text{ included } \text{Node.all} \text{ and } h_4 \text{ included } h_1 \].

### 5.3.2 Graph Modelling

At the beginning of when starting the network, every node will send hello message in the Initial state. Transition SendingHLL will receive input token from initial place, Connected place, Actived place. Connected place will help to identify connecting between these node for sending output to 1HopIdentify place. The output to 1HopIdentify is HelloMessage, that will generate by identify the connect between 2 hop \( x \) and \( y \); \( x \) is the node that sending hello message and their 1 hop list, and \( y \) is the destination node that
will receive the hello messages of \( x \). Another place will receive output from SendingHLL transition is Actived place, this place will receive all activated node and their 1 and 2 hop connecting list.

At ReceiveHLL Transition, this state will mainly simulate the process of finding 1 and 2 hop connecting node at each node in the network. Transition will be fired when receive input source from Actived, Connode and 1HopIdentify places. At the arc send out put token return to Actived Place, List 1 hop of \( y \) Node will add node which is \( x \) sending hello message to, and with list 2 hop will add list 1 hop of sending hellomessage node \( x \), Syntax of the arc is: \(< y, \{\{h3\} \cup \{x\}\}, \{h4\}\cup h1, h1 \>>\). Output token which is send to 1HopIdentify will update the hello-message of \( y \) node to the destination \( z \), which is identify by receive the input token from Connode to get all the connecting node to \( y \).

At the final of this initial state, FinishInit transition will guard for checking condition of every node for ending send hellomessage state of the node. This transition will get input token from Actived place and 1HopIdentify. At 1HopIdentify place, information of all hello message of node which is out put from Actived place will be checked. If there are no change between hello messgae and 2 list of node, this node will end its initial state and no more sending hello message. With the guard \([h3 \text{ included } \text{Node.all}]\) will checked there is no more node will send hello message to node \( y \), and the another guard \([h4 \text{ included } h1]\) will checked that in hello message there is no more node that need to add to \( h4 \) of \( y \). After all checked, transition will send the out put to ActivedNodeComplete.

6 Conclusion and future works

6.1 Conclusion

During 5 months working in the lab, I have chance to research, implement a model to generate modelling on Petri net. The model is correct on the syntax and it have an agree from my professors. But there are an issue on the Modelling tools, that alligator tool for checking model is not works. I have contact with mr. Fabrice Kordon for checking and fixing the problems. When this fixing finishes, it can check for correct the model and apply on the real model.

After my work, we can continue extend the work on next states for archiving the goal of project. The model on initial state is agreed by my professors, and it will continue to work on. The model simulate correctly the processing of initial state on WSNs. It can easily for extending on the next steps.

6.2 Future work

In the future, this project will continue to work for:

- Using Alligator tools for checking model on initial state that be made in section 5.
- Apply a real example on model for checking the right on large scale of the network.
• Extending the next state of this network: Sink join the network, Sink change position, Local repair.... These step must be work step by step on this project.

• Giving the result for energy effective of WSNs by checking model on Cosyverif and Alligator tools.

References


