Probabilistic Congestion of Wireless Sensor Networks: a Coloured Petri Net based Approach

Khanh Le
Bach Khoa University, Vietnam

Thanh Cao
Saigon University, Vietnam

Phuc Le
Bach Khoa University, Vietnam

Bao Pham
Bach Khoa University, Vietnam

Thang Bui
Bach Khoa University, Vietnam

Tho Quan
Bach Khoa University, Vietnam

ABSTRACT

Analysing probability properties on Coloured Petri Nets (CPNs) model is one of a favorite topic on system verification recently. This paper focuses on verifying congestion probability on Wireless Sensor Networks (WSNs) which is modelled by CPN. Actually, WSNs are the collection of sensors. A WSN topology is formed by the interaction among sensors via Wi-Fi connections. However, sensors can be consider as unsteady devices when working in the harsh environment due to limited processing capacity, non-replacement battery, etc. Hence, each sensor needs to attach a reliable probability so that users can know the probability of reaching the sink of data. Such probabilities are added into the transitions in our CPN probability model before checking congestion. Whole verifying process introduces also in order to emphasize the purpose of this paper via a straight example.

General Terms

System Modelling, System Verification

Keywords

Wireless Sensor networks, Realiable Probabilisties, Congestion detection, Concurrency architecture, Petri nets, Coloured Petri nets

1. INTRODUCTION

Wireless Sensor Networks (WSNs) consist a hundred or a thousand of sensors that collect data from the environment and send them to the sink. Sensor nodes which are limited resource devices, communicate with each others through radio channels. WSNs can be deployed in many harsh environment such as in the jungle or under the ocean for habitat monitoring [24]. WSNs need to solve many unwanted factors when working in these applications:

(1) Sensors are powered with limited battery and thus can not be easily replaced when failed or exhausted in unpleasant environment. Moreover, they can be consider as low processing capacity as well.
(2) Data signal is easily interference in such region.

These reasons lead to congestion on WSNs. Congestion means that data from the source sensor is dropped somewhere in the transmission path and thus it cannot reach to sink node [1].

In recent years, there have been a number of approaches of congestion control for WSNs [21, 6, 22]. Congestion control is divided two steps including *congestion detection* and *congestion mitigation*. In this paper, we focus on the first one. In turn, there are two main reasons of congestion in WSN [22]. The first reason is *node-level congestion*. When the rate of incoming packets to a node is faster than the service's rate, it causes buffer overflow. The second reason is *link-level congestion*. Collision could occur when multiple sensors try to send data to the channel at the same time.

Obviously, it is important to control the congestion in WSNs. One way to detection the congestion in WSN is doing experiments with *simulation* and the other way is analysing with *formal modelling*. In the first case, a simulator is used to mimic the operations of the WSN, measure the performance, and check whether a certain anomaly like congestion occurs or not. Widely used simulators include ns2 [16] and Omnet++ [19]. In these, a WSN is mostly considered as a network with sensors, channels and their activity (protocols). The activity of protocols in network is programmed based on the help of simulator frameworks. Hence, users must program their models according to the protocol used. The second case appears with two immediate advantages, comparing to simulator approaches:

(1) the WSN is modelled at a higher level of abstraction, only including sensors and channels. Thus, the WSN model is independent of the framework used.
(2) the model defines all scenarios and allows for exhaustively model checking desired properties.

Another noticed point concerns to sensor's activities, the sensor node can execute a number of concurrent operations such as generating packets (capturing information), processing, and sending packets; while the channel concurrently receives and transmits packets [2]. Moreover, a component can operate independently with each others. Therefore, operations of sensors and channels in a WSN should be performed concurrently, i.e. all enabled transitions in the modelled-PN of the WSN should fire simultaneously if they are no conflict. This is called *concurrency* of WSN which expresses properly the ideal operation of a WSN.

The proposal of [9] proposes a congestion-based method based on the modelling approach, called *WSN-PN*. This model is created by using P/T Petri nets and only focuses on the distribution of packets of a WSN. The moving of tokens when fired represents packet's movement. Furthermore, *WSN-PN* works based on sequential semantic (i.e. only one enabled transition can fire at a time) on analysing. That can lead deviated diagnoses since in a WSN, its components can operate concurrently at a time [5].

This paper is constructed the ideas based on WSN's environment applications, i.e. all devices have reliable working probability. The main raising problem here is all components of WSN cannot operate correctly due to some constraints such as environmental interference, low power in sensor node, etc. Thus, attaching reliable probability on devices increases advantage when system verification. There are currently plenty of works on WSN reliability modelling and reliability analysis [23, 20]. Most of such works is applied to the developing countries with low infrastructure or in extremely harsh environment and hostile region [14, 13].

To the best of our knowledge, [10] is the sole work that verifying the congestion probability based on Discrete Time Stochastic

Petri Nets so far. This proposed model called WDSTPN, mentions about reliable probability of WSN's components. So, it can imply the probability of reaching congestion of a WSN model instead of answering whether congestion happen or not. Even though it is modelled by P/T Nets, the sensors on network still perform their task under concurrency semantic, the semantics of WSN's operations must be expresses by using code program associated to transitions. These code are in form of C# language, thus forcing the human modeller must get meanings of both PN and C# language. Moreover, the human modeller also difficultly maps components of the modelled PN to corresponding parts of these code. The main cause of the disadvantage is the limitations of P/T nets and corresponding supporting tools. This urges us to consider other advanced kinds of PN, Coloured Petri nets (CPNs) [8, 7]. Using CPNs can overcome the disadvantage specifying as follows. First, parameters of a sensor can be expressed through a colour set. Second, CPNs support arc expressions which can include constants, variables, and functions. Therefore, it is possible to express explicitly the semantics of WSN operations on the modelled CPN. The human modeller can only concentrate on functions (in form of CPN language) to understand and check whether these functions express properly the semantics of WSN operations.

In the following sections, the approach of this proposal is a probabilistic model for congestion detection on WSNs by using CPNs. The work helps to express a WSN more rational in actual context by reusing way to attach reliable probability on WSN network topology from [10] and try to transform this topology into WDTSPN. Finally, we find a new technique to detect congestion on this model.

Outline. The rest of the paper is organised as follows. Section 2 gives overview of Coloured Petri nets and the model of WSN based on Coloured Petri nets. Section 3 defines a probabilistic wireless sensor network with reliable and routing probabilities. Then, Section 4 defines a probabilistic model of WSN based on Discrete Time Stochastic Petri Nets. More extensive analysis on our model is reported in Section 5. We also emphasize the technique to detect congestion in Section 6. Finally, 7 draws conclusions and outlines future work.

2. PRELIMINARIES

2.1 Coloured Petri Nets

Coloured Petri nets (CPNs for short) [8, 12, 7] is a modelling language which combines the strengths of Petri nets with the expressive power of functional programming languages. In CPNs, tokens are distinguished by the "colour" instead of only the "black" one. Moreover, arc expressions (an extended version of arc weights in classical Petri nets) specify which tokens can flow over the arcs. Guards that are Boolean expressions defining additional constraints on enabling transitions.

Let EXP be the set of expressions complying with the CPN-ML [3] syntax. EXP is used to express components of CPN such as arc expressions, guards. Before expressions are evaluated to values, the variables in the expressions must be assigned values, which is called *binding*. Besides, a multiset is a set in which there can be several occurrences of the same token. Multiset is defined in Definition 1 [7]. It is an important concept used in the later definitions of markings, steps, and the enabling and occurrence of transitions. Definition 2 [7] gives the formal definition of CPNs.

DEFINITION 1. Let $S = \{s_1, s_2, ...\}$ be a non-empty set. A *multiset* over S is a function $m : S \to \mathbb{N}$ that maps each element $s \in S$ into a non-negative integer $m(s) \in \mathbb{N}$ called the *number of appearances* (coefficient) of s in m. A multiset m can also be

written as a sum:

$$^{++}\sum_{s \in S} m(s)`s = m(s1)`s1 + + m(s2)`s2 + + ...$$

Membership, *addition*, *comparison*, and *subtraction* are defined as follows, where m_1, m_2, and m are multisets:

1. $\forall s \in S : s \in m \Leftrightarrow m(s) > 0$
2. $\forall s \in S : (m_1 + + m_2)(s) = m_1(s) + m_2(s)$
3. $m_1 <<= m_2 \Leftrightarrow \forall s \in S : m_1(s) \leq m_2(s)$
4. When $m_1 <<= m_2$,
$\forall s \in S : (m_2 - - m_1)(s) = m_2(s) - m_1(s)$

The set of all multisets over S is denoted S_{MS}.

DEFINITION 2. A *Coloured Petri net* is a nine-tuple $\langle P, T, F, \Sigma, V, c, g, f, m_0 \rangle$ where:

1. P is a finite, nonempty set of *places*.
2. T is a finite, nonempty set of *transitions* such that $P \cap T = \emptyset$.
3. $F \subseteq P \times T \cup T \times P$ is a finite set of directed *arcs*.
4. Σ is a finite set of non-empty *colour sets*.
5. V is a finite set of *typed variables* such that $Type[v] \in \Sigma$ for all $v \in V$.
6. $c : P \to \Sigma$ is a *colour set function* that assigns to each place $p \in P$ a colour set $c(p) \in \Sigma$.
7. $g : T \to EXP$ is a *guard function* that assigns to each transition $t \in T$ a guard expression of the Boolean type.
8. $f : F \to EXP$ is an *arc expression function* that assigns to each arc $a \in F$ an arc expression of a multiset type $c(p)_{MS}$, where p is the place connected to the arc a.
9. $m_0 : P \to EXP$ is an *initialisation function* that assigns to each place $p \in P$ an initialisation expression of a multiset type $c(p)_{MS}$.

DEFINITION 3. For a Coloured Petri net $CPN = \langle P, T, F, \Sigma, c, g, f, m_0 \rangle$, defining the following concepts:

1. A *marking* is a function M that maps each place $p \in P$ into a multiset of tokens where $M(p) \in C(p)_{MS}$. m_0 is *initial marking* of CPN.
2. The *variables of a transition* t are denoted $Var(t) \in V$ and consit of the free variables appearing in guard of t and in the arc expressions of arcs connected to t.
3. A *binding* of a transition t is a function b that maps each variable $v \in Var(t)$ into a value $b(v) \in Type[v]$. The set of all bindings for a transition t is denoted $B(t)$.
4. A *binding element* is a pair (t, b) such that $t \in T$ and $b \in B(t)$. The set of all binding elements $BE(t)$ for a transition t is defined by $BE(t) = \{(t, b)|b \in B(t)\}$. The set of all binding elements in a CPN model is denoted BE.
5. A *step* $Y \in BE_{MS}$ is a non-empty, finite multiset of binding elements.

Definition 3 [7] defines concepts used to express the semantics of CPNs. Then the enabling and occurrence of a binding element are summarised in Definition 4 [7].

DEFINITION 4. A binding element $(t, b) \in BE$ is *enabled* in a marking M if and only if the following two properties are satisfied:

1. $G(t)\langle b \rangle = true$
2. $\forall p \in P : E(p, t)\langle b \rangle <<= M(p)$

When (t, b) is *enabled* in M, it may occur, leading to the marking M' defined by:

3. $\forall p \in P : M'(p) = (M(p) - - E(p, t)\langle b \rangle) ++ E(t, p)\langle b \rangle$

M' is directly reachable from M. This is denoted by $M \xrightarrow{(t,b)} M'$. The set of markings reachable from a marking M is denoted $\mathfrak{R}(M)$.

A CPN model can be organised as a set of hierarchically related modules. CPN models with modules are also called *hierarchical Coloured Petri Nets* (HCPNs for short) [8, 7]. A HCPN allow dividing a module into smaller modules (sub modules) connected to each other using well-defined interfaces (substitution transitions and fusion places). With HCPNs, human modeller can work at different abstraction levels and concentrate on only a few details at a time. Moreover, a module is defined once and is used repeatedly. This allows reading only one description, and to modify one description when changes are necessary.

2.2 CPN Model for WSN

[18] proposes a CPN model for WSN called WSN-CPN. This model represents the operations of each WSN component: sensors and channels. The modelled operations includes: generating packets of source sensors, internal processing of intermediate and sink sensors, sending/receiving packets of sensors/channels, and transmitting packets of channels.

The WSN-CPN model is composed of seven modules. Figure 1 shows module hierary of the proposed model. The "Top" module is prime module with no incoming arc. It is decomposed into two sub-modules "Initialisation" and "Processing". The "Initialisation" module is responsible to initialise markings for working places of the overall model. The "Processing" module includes four sub-modules such as "Generate Packet", "Internal Process", "Receive Packet", and "Transmit Packet" corresponding to main operations of all WSN components. See [18] for more details.

Fig. 1: Module hierarchy of the proposed model.

The structure of the WSN-CPN model is fixed. When the topology changes, only initial marking needs changed respectively. In this paper, a simple example WSN depicted in Figure 2 is used as a straight case study. Table 1 gives parameter configuration of the WSN. Listing 1 expresses value of constant "iniNetwork" assigned to initial marking of place "network" of the `WSN-CPN` model of WSN in Figure 2. Assume that unicat mode is considered.
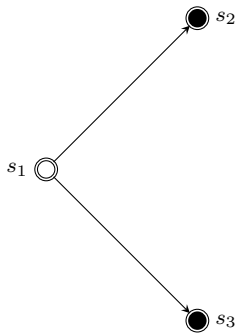

Fig. 2: Topology of the example WSN with three sensors including one source sensor s_1 (denoted by a double-lined circle), two sink sensors s_2 and s_3 (denoted by two full circles).

Table 1. : The parameter configuration of the WSN in Figure 2

	Source	Sink	Intermediate	Channel
Sending rate	3	3	3	N/A
Buffer size	5	5	5	5
Queue size	5	5	5	N/A
Processing rate	5	5	5	N/A
Transmission rate	N/A	N/A	N/A	3
Number of packets	6			

Listing 1: Initial marking of the "network" place.

```
val iniSensors = [(1,{ typ = Source, buf = 0, queue = 0,
  pmax = 6, sending_rate = 3, processing_rate = 5,
  buf_size = 5, queue_size = 5}),(2,{ typ =
  Sink, buf = 0, queue = 0, pmax = 0,
  sending_rate = 3, processing_rate = 5, buf_size = 5,
  queue_size = 5}),(3,{ typ = Sink, buf = 0,
  queue = 0, pmax = 0, sending_rate = 3,
  processing_rate = 5, buf_size = 5, queue_size = 5})];

val iniChannels = [(1,[({ channel_buf = 0, trans_rate =
  3, channel_buf_size = 5},2)],1),(1,[({ channel_buf =
  0, trans_rate = 3, channel_buf_size = 5},3)],2)];

val iniChannelMode = Unicast;

val iniNetwork = { channel_mode = iniChannelMode,
  net_sensors = iniSensors, net_channels = iniChannels}
```

3. PROBABILISTIC WIRELESS SENSOR NETWORKS

Reliability of a system is a probability that it operates correctly [4]. In WSN context, every component in a WSN has a reliable probability [23]. The failure (unreliability) probability of a component is defined as the subtraction of 1 and its reliable probability. This research only focuses on the functional aspect of reliability. Thus, the reliable probability of a sensor or a channel is regarded as a correcting functional probability.

Following [9], in a WSN, a source sensor has two functions including generating and sending packets; a sink sensor has processing packets function; and an intermediate two functions including processing and sending packets. We define some reliable probabilities (correcting functional probabilities) for components of a WSN as following:

— λ_{sg} be the reliable probability for generating packets function of a source sensor.
— λ_{ss} be the reliable probability for sending packets function of a source or an intermediate sensor.
— λ_{sp} be the reliable probability for processing packets function of a sink sensor and an intermediate sensor.
— λ_{ct} be the reliable probability for transmitting packets function of a channel.

In the unicast mode, if a sensor operates correctly, it must choose only one of its neighbour sensors to send packets. The chosen probabilities are different and defined as *routing probabilities* [17]. This research assumes that routing probability of each sensor has been specified initially.

Definition 5 defines a *Probabilistic Wireless Sensor Network* (PWSN) combining a WSN with reliable probabilities and routing probabilities.

DEFINITION 5 PROBABILISTIC WIRELESS SENSOR NETWORK. A *Probabilistic Wireless Sensor Network* (PWSN) is a seven-tuple $\langle S_{source} \cup S_{intermediate} \cup S_{sink}, C, \lambda_{sg} \cup \lambda_{ss} \cup \lambda_{sp} \cup \lambda_{ct}, \alpha_{sr} \rangle$ where:

1. S_{source} is a finite, nonempty set of *source sensors*.
2. $S_{intermediate}$ is a finite set of *intermediate sensors*.
3. S_{sink} is a finite, nonempty set of *sink sensors*.
4. $C = \bigcup_{i=\overline{1,n}} C_i$ is a finite, nonempty set of *channels*. Herein, $C_i \cap C_j = \emptyset$ with $\forall i, j \in \{1, ..., n\}; i \neq j$. Each C_i represents for a set of all channels which have the same from-sensor.
5. $\lambda_{sg} : S_{source} \to \mathbb{R}$ is a *probabilistic function* assigning to each sensor $s \in S_{source}$ a probability $p \in \mathbb{R}$. p represents the reliable probability for generating packets function of s.
6. $\lambda_{ss} : S_{source} \cup S_{intermediate} \to \mathbb{R}$ is a *probabilistic function* assigning to each sensor $s \in S_{source} \cup S_{intermediate}$ a

probability $p \in \mathbb{R}$. p represents the reliable probability for sending packets function of s.

7. $\lambda_{sp} : S_{intermediate} \cup S_{sink} \to \mathbb{R}$ is a *probabilistic function* assigning to each sensor $s \in S_{intermediate} \cup S_{sink}$ a probability $p \in \mathbb{R}$. p represents the reliable probability for internal processing packets function of s.

8. $\lambda_{ct} : C \to \mathbb{R}$ is a *probabilistic function* assigning to each channel $c \in C$ a probability $p \in \mathbb{R}$. p represents the reliable probability for transmitting packets function of c.

9. $\alpha_{sr} : C \to \mathbb{R}$ is a *probabilistic function* assigning to each channel $c \in C$ a probability $p \in \mathbb{R}$. p represents the routing probability for sending packets function of from-sensor of c. If the WSN is in broadcast mode, the following equation must be satisfied.

$$\alpha_{sr}(c) = 1.0, \quad \forall c \in C \tag{1}$$

If the WSN is in unicast mode, the following equation must be satisfied.

$$\sum_{c \in C_i} \alpha_{sr}(c) = 1.0, \quad \forall i \in \{1, ..., n\} \tag{2}$$

For example, with WSN in Figure 2 we have a following PWSN based on Definition 5:

— $S_{source} = \{s_1\}$

— $S_{intermediate} = \emptyset$

— $S_{sink} = \{s_2, s_3\}$

— $C = C_1 = \{c_{1-2}, c_{1-3}\}$, the from-sensor of all channels in C_1 is s_1

— $\lambda_{sg}(s_1) = \lambda_{ss}(s_1) = 0.999$ (high reliabilities)

— $\lambda_{sp}(s_2) = \lambda_{sp}(s_3) = 0.999$ (high reliabilities)

— $\lambda_{ct}(c_{1-2}) = \lambda_{ct}(c_{1-3}) = 0.999$ (high reliabilities)

— $\alpha_{sr}(c_{1-2}) = \alpha_{sr}(c_{1-3}) = 0.5$ (considering in unicast mode)

4. DISCRETE TIME STOCHASTIC COLOURED PETRI NET MODEL OF WSN

Most papers focus on using Markov model to express reliability of WSNs, for instance in [20, 23]. However, this paper uses Discrete Time Stochastic Coloured Petri Net (DTSCPN), an extension with colour of Discrete Time Stochastic Petri Net, to model WSN with the reliable probabilities due to two reasons. First, Discrete Time Stochastic Petri Net (DTSPN) is a powerful technique to analyse stochastic aspects [11]. DTSPN is more observable, scalable, and portable than Markov model [23]. Second, DTSPN allows that more than one transition (binding element) may fire (occur) at a time step. This helps to express the concurrency of WSN. Additionally, we combine routing probabilities with reliable probabilities in our DTSCPN model.

In theory of DTSCPNs, each binding element of a DTSCPN has a non-zero conditional probability. A DTSCPN is extended from a DTSPN by adding the set of non-zero conditional probabilities $\rho = \{\rho_1, \rho_2, ..., \rho_m\}$ to binding elements instead of transitions. The probability $\rho_i < 1.0$ denotes the probability that the enabled binding element t_i occurs at the next time step, given (conditioned on) the fact that no other binding element occurs. The non-zero conditional probabilities is fixed and initial specified. The DTSCPN model of our WSN, called *WDTSCPN*, is defined in Definition 6.

DEFINITION 6 WDTSCPN MODEL. Let $\mathcal{C} = \langle P, T, F, \Sigma, c, g, f, m_0 \rangle$ be CPN model of PWSN $\mathcal{W} = \langle S_{source} \cup S_{intermediate} \cup S_{sink}, C, \lambda_{sg} \cup \lambda_{ss} \cup \lambda_{sp} \cup \lambda_{ct}, \alpha_{sr} \rangle$. Let BE be the set of all binding elements in $\mathcal{C}$. *WDTSCPN* is a teen-tuple $\langle P, T, F, \Sigma, c, g, f, m_0, \rho \rangle$ where:

1. $\rho : BE \to \mathbb{R}$ is a *probabilistic function* assigning each binding element $eb \in BE$ a probability $p \in \mathbb{R}$;

2. If $eb = (\text{generate packet}, b)$ and $b(s\_id) = s$ then $\rho(eb) = \lambda_{sg}(s)$;

3. If $eb = (\text{internal process}, b)$ and $b(s\_id) = s$ then $\rho(eb) = \lambda_{sp}(s)$;

4. If $eb = (\text{receive packet}, b)$ and $b(from\_s\_id) = s, b(c\_id) = c$ then $\rho(eb) = \lambda_{ss}(s) \times \alpha_{sr}(c)$;

4. If $eb = (\text{transmit packet}, b)$ and $b(c\_id) = c$ then $\rho(eb) = \lambda_{ct}(c)$.

Considering WSN in Figure 2 and following Definition 6, we have:

— $gen_1 = (\text{generate packet}, \langle s\_id = 1, s\_param = \{typ = Source, buf = 0, queue = 0, pmax = 6, sending\_rate = 3, processing\_rate = 5, buf\_size = 5, queue\_size = 5\}\rangle)$ then $\rho(gen_1) = \lambda_{sg}(1)$

— $rec_{1-2} = (\text{receive packet}, \langle s\_id = 1, s\_param = \{typ = Source, buf = 0, queue = 3, pmax = 3, sending\_rate = 3, processing\_rate = 5, buf\_size = 5, queue\_size = 5\}, c\_list = [(\{channel\_buf = 0, trans\_rate = 3, channel\_buf\_size = 5\}, 1)], c\_id = 1 \rangle)$ then $\rho(rec_{1-2}) = \lambda_{ss}(1) \times \alpha_{sr}(1)$

5. NEW ANALYSIS ALGORITHM FOR WDTSCPN MODEL

In theory of DTSCPNs, all enabled binding elements in a marking are split into group of individual binding elements (not conflict) and group of binding elements which conflict with each other. The actions of the binding elements in each group are independent of binding elements in the other groups, and can be solved separately [15]. However, in unicast mode, a sensor must choose only one of its neighbour sensors to send packets, i.e. in the WSN-CPN model, although binding elements rec_{1-2} and rec_{1-3} are enabled and not conflict by theory of DTSCPN, they cannot be in non-conflict group to obey the constraint of unicast mode. This urges us to propose a new analysis algorithm for WDTSCPN model.

Next, defining a mutually exclusive set called *unicast-set* in Definition 7, which reflexes such constraint.

DEFINITION 7 UNICAST-SET. Let BE be set of enabled binding elements of the WDSTCPN model at marking M. An *unicast-set* U is a set of all enabled binding elements $\{(t_1, b_1), ..., (t_m, b_m)\}; m \geq 1$ such that:

1. $U \subseteq BE$;
2. $t_i = \text{receive packet}$ with $\forall i \in \{1, 2, .., m\}$;
3. $b_i(from\_s\_id) = b_j(from\_s\_id)$ with $\forall i, j \in \{1, 2, .., m\}$;
4. and U is a mutually exclusive set, i.e. the occurring of any subset of binding elements $U' \subset U$ results in a marking in which some other binding element $(t_j, b_j) \notin U'$ is disabled.

Since an unicast-set U is a mutually exclusive set, the event space for the enabled binding elements of U in the marking M is of size $|U| + 1$. Let E_i be the event that binding element $eb_i \in U$ occurs and let E_0 be the event that no binding element of U fires. Since no two binding elements can occur at the same time in this set, the events are mutually exclusive and the probabilities must sum to 1. Since the nonzero conditional probabilities of the binding elements in a unicast-set are dependent and conditional to each other, we cannot apply the calculation method based on conditional probability of original DTSPN. Instead, we can conclude that the probability that a particular binding element in an
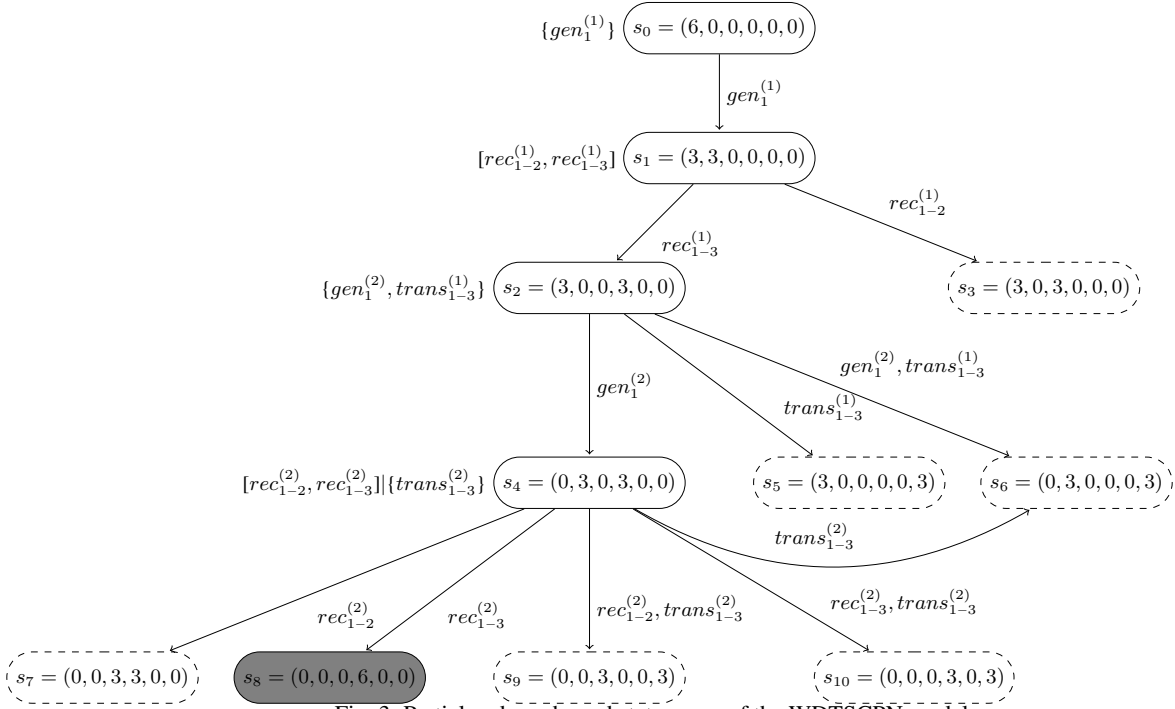
Fig. 3: Partial and condensed state space of the WDTSCPN model.

unicast-set set will occur is

$$P[E_i] = \rho(eb_i)$$

and the probability that no binding element in an unicast-set set will occur is

$$P[E_0] = 1 - \sum_{1}^{n} \rho(eb_i).$$

The state space with transition probabilities of WDTSCPN model is calculated following the theory of DTSPN [15]. Then, we define a new probabilistic analysis method on the calculated state space. Furthermore, the method is applied on congestion detection problem of WSNs.

DEFINITION 8 TRANSITION PROBABILITY. Let ec be a set of combined binding elements of state (marking) a and $a \xrightarrow{ec} b$, then $P[a \rightarrow b] = P[ec$ occured$]$ is the transition probability from state a to state b. $P[ec$ occured$]$ is calculated by theory of DTSPNs [15].

DEFINITION 9 SEQUENCE PROBABILITY. Let $sq = s_0 \rightarrow s_1 \rightarrow ... \rightarrow s_n \rightarrow s$ be sequence of states from s_0 to s. The probability of this sequence is defined as

$$P[sq] = P[s_0 \rightarrow s_1] \times ... \times P[s_n \rightarrow s] \qquad (3)$$

DEFINITION 10 REACHING PROBABILITY. Let $P(s)$ be probability of reaching to state s from initial state s_0 (initial marking m_0). Let $\mathbb{S}$ be a set of all sequences of states from s_0 to s. We have

$$P(s) = \sum_{sq \in \mathbb{S}} P[sq] \qquad (4)$$

Figure 3 gives a partial and condensed state space of the WDTSCPN model of the WSN in unicast mode of Figure 2. A marking (node) is stood by a rounded rectangle and expresses the value of six-tuple $(p, q_1, cb_{1-2}, cb_{1-3}, b_2, b_3)$. Herein, p denotes number of packets, q_1 denotes number of current packets in queue of Sensor s_1, cb_{1-2} denotes number of current packets in buffer of Channel $s_1 - s_2$, cb_{1-3} denotes number of current packets in buffer of Channel $s_1 - s_3$, b_2 denotes number of packets,

q_1 denotes number of current packets in buffer of Sensor s_2, b_3 denotes number of packets, q_1 denotes number of current packets in buffer of Sensor s_3. The partition set of binding elemtns of a node is shown in left of the node. For instance, $[rec_{1-2}, rec_{1-3}]$ is an unicast-set, $\{trans_{1-3}\}$ is a no-conflict set in group of individual no-conflict binding elements. Each arc from node a to node b is labelled by a list of occured binding elements making marking a transit to marking b. The node, which are denoted by a dashed and rounded rectangle, is a folding node (i.e. not be expanded in the state space). The grey node is where occurs congestion. There are list of some binding elements appearing in Figure 3:

— $gen_1^{(1)} = $ (generate packet, $\langle s\_id = 1, s\_param = \{typ = Source, buf = 0, queue = 0, pmax = 6, sending\_rate = 3, processing\_rate = 5, buf\_size = 5, queue\_size = 5\}\rangle)$

— $rec_{1-2}^{(1)} = $ (receive packet, $\langle s\_id = 1, s\_param = \{typ = Source, buf = 0, queue = 3, pmax = 3, sending\_rate = 3, processing\_rate = 5, buf\_size = 5, queue\_size = 5\}, c\_list = [(\{channel\_buf = 0, trans\_rate = 3, channel\_buf\_size = 5\}, 1)], c\_id = 1\rangle)$

— $rec_{1-3}^{(1)} = $ (receive packet, $\langle s\_id = 1, s\_param = \{typ = Source, buf = 0, queue = 3, pmax = 3, sending\_rate = 3, processing\_rate = 5, buf\_size = 5, queue\_size = 5\}, c\_list = [(\{channel\_buf = 0, trans\_rate = 3, channel\_buf\_size = 5\}, 2)], c\_id = 2\rangle)$

— $trans_{1-3}^{(1)} = $ (transmit packet, $\langle to\_s\_id = 3, s\_param = \{typ = Source, buf = 0, queue = 0, pmax = 3, sending\_rate = 3, processing\_rate = 5, buf\_size = 5, queue\_size = 5\}, from\_s\_id = 1, c\_list = [(\{channel\_buf = 3, trans\_rate = 3, channel\_buf\_size = 5\}, 2)], c\_id = 2\rangle)$

We have

$$
\begin{aligned}
P(s_2) \\
&= P[s_0 \to s_1 \to s_2] \\
&= P[s_0 \to s_1] \times P[s_1 \to s_2] \\
&= P[gen_1^{(1)} \text{ occured}] \times P[rec_{1-3}^{(1)} \text{ occured}] \\
&= \rho(gen_1^{(1)}) \times \rho(rec_{1-3}^{(1)}) \\
&= \lambda_{sg}(1) \times \lambda_{ss}(1) \times \alpha_{sr}(2) \\
&= \lambda_{sg}(s_1) \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \\
&= 0.999 \times 0.999 \times 0.5 \\
&= 0.499
\end{aligned}
$$

and

$$
\begin{aligned}
P(s_6) \\
&= P[s_0 \to s_1 \to s_2 \to s_6] + P[s_0 \to s_1 \to s_2 \to s_4 \to s_6] \\
P[s_0 &\to s_1 \to s_2 \to s_6] \\
&= P[s_0 \to s_1] \times P[s_1 \to s_2] \times P[s_2 \to s_6] \\
&= P(s_2) \times P[s_2 \to s_6] \\
&= P(s_2) \times P[gen_1^{(2)}, trans_{1-3}^{(1)} \text{ occured}] \\
&= P(s_2) \times \rho(gen_1^{(2)}) \times \rho(trans_{1-3}^{(1)}) \\
&= P(s_2) \times \lambda_{sg}(1) \times \lambda_{ct}(2) \\
&= P(s_2) \times \lambda_{sg}(s_1) \times \lambda_{ct}(c_{1-3}) \\
&= \lambda_{sg}(s_1) \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \times \lambda_{sg}(s_1) \times \lambda_{ct}(c_{1-3}) \\
&= \lambda_{sg}(s_1)^2 \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \times \lambda_{ct}(c_{1-3}) \\
&= 0.999^2 \times 0.999 \times 0.5 \times 0.999 \\
&= 0.498 \\
P[s_0 &\to s_1 \to s_2 \to s_4 \to s_6] \\
&= P(s_2) \times P[s_2 \to s_4] \times P[s_4 \to s_6] \\
&= P(s_2) \times P[gen_1^{(2)} \text{ occured}] \times P[trans_{1-3}^{(2)} \text{ occured}] \\
&= P(s_2) \times (\rho(gen_1^{(2)}) \times (1 - \rho(trans_{1-3}^{(1)}))) \times \\
& \quad (\rho(trans_{1-3}^{(2)}) \times (1 - \rho(rec_{1-2}^{(2)}) - \rho(rec_{1-3}^{(2)}))) \\
&= P(s_2) \times \lambda_{sg}(1) \times (1 - \lambda_{ct}(2)) \times \\
& \quad \lambda_{ct}(2) \times (1 - \lambda_{ss}(1)) \\
&= P(s_2) \times \lambda_{sg}(s_1) \times (1 - \lambda_{ct}(c_{1-3})) \times \\
& \quad \lambda_{ct}(c_{1-3}) \times (1 - \lambda_{ss}(s_1)) \\
&= \lambda_{sg}(s_1) \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \times \lambda_{sg}(s_1) \times \\
& \quad (1 - \lambda_{ct}(c_{1-3})) \times \lambda_{ct}(c_{1-3}) \times (1 - \lambda_{ss}(s_1)) \\
&= \lambda_{sg}(s_1)^2 \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \times \\
& \quad (1 - \lambda_{ct}(c_{1-3})) \times \lambda_{ct}(c_{1-3}) \times (1 - \lambda_{ss}(s_1)) \\
&= 0.999^2 \times 0.999 \times 0.5 \times \\
& \quad (1 - 0.999) \times 0.999 \times (1 - 0.999) \\
&= 0.498e^{-6} \\
P(s_6) &= 0.498 + 0.498e^{-6} = 0.498
\end{aligned}
$$

6. PROBABILISTIC CONGESTION OF WIRELESS SENSOR NETWORKS

DEFINITION 11. *Probabilistic Congestion* A probabilistic congestion is a two-tuple (s, p) where:

1. s is a state (marking) which occurs congestion (i.e. its current packets in buffer exceed the buffer size).
2. $p = P(s)$

In Figure 3, s_8 is a congestion state. We have

$$
\begin{aligned}
P(s_8) &= P[s_0 \to s_1 \to s_2 \to s_4 \to s_8] \\
&= P(s_2) \times P[s_2 \to s_4] \times P[s_4 \to s_8] \\
&= P(s_2) \times P[gen_1^{(2)} \text{ occured}] \times P[rec_{1-3}^{(2)} \text{ occured}] \\
&= P(s_2) \times (\rho(gen_1^{(2)}) \times (1 - \rho(trans_{1-3}))) \times \\
& \quad (\rho(rec_{1-3}^{(2)}) \times (1 - \rho(trans_{1-3}^{(2)}))) \\
&= P(s_2) \times \lambda_{sg}(1) \times \\
& \quad (1 - \lambda_{ct}(2))^2 \times \lambda_{ss}(1) \times \alpha_{sr}(2) \\
&= P(s_2) \times \lambda_{sg}(s_1) \times \\
& \quad (1 - \lambda_{ct}(c_{1-3}))^2 \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \\
&= \lambda_{sg}(s_1) \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \times \lambda_{sg}(s_1) \times \\
& \quad (1 - \lambda_{ct}(c_{1-3}))^2 \times \lambda_{ss}(s_1) \times \alpha_{sr}(c_{1-3}) \\
&= \lambda_{sg}(s_1)^2 \times \lambda_{ss}(s_1)^2 \times \alpha_{sr}(c_{1-3})^2 \times (1 - \lambda_{ct}(c_{1-3}))^2 \\
&= 0.999^2 \times 0.999^2 \times 0.5^2 \times (1 - 0.999)^2 \\
&= 0.249e^{-6}
\end{aligned}
$$

We have a probabilistic congestion $(s_8, 0.249e^{-6})$. State s_6 (indicating no occurring congestion) is matched with the concurrency of WSN while state s_8 is not. $P(s_8) \ll P(s_6)$ ($0.249e^{-6} \ll 0.498$). The probability of occurring congestion is very small against the probability of no occurring congestion. That reflexes properly the actual operation of a WSN.

7. CONCLUSION

As discussed, DTSCPN is an extension of DTSPN which is considered as a powerful technique to analyse stochastic aspects. In addition, using CPN to accomplish a WSN model in congestion detection context can efficiently express all its activities including concurrency side. This paper has proposed the WDTSCPN model that is the combination result of PN-Modelled by CPN and verification by DTSPN. The pair of reliable and choosing path probabilities are added into transitions of model to check the probability of reaching congestion on WSNs. Thus, proposing new analysis technique is also an emphasized point in this research. Straight example is an illustration for our propose.

In reality, sensor's reliability is also depending on several constraints such as power availability, sensor's position (if the nodes are mobile), status of sensors (i.e. activation or sleeping) etc. The more level of power consumption, the more reliable processing on sensor nodes. Moreover, sensor moving causes changing topology and they will spend energy to re-localize themselves. Therefore, in the future, sensor's reliability should be represented like a dynamic parameter in WDTSCPN model. This leads to define a new way to attach and control such variable. the future, we will apply such model in the real context.

8. REFERENCES

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[2] Zaheer Aslam, Nauman Qamar, Shehzad Rizwan Noor Khan, Kamal Ahmad, Noor Zaman, Shafiullah, and Rashid Zubair. A survey of wireless sensor network-software architecture design issues. *International Journal of Computer Science and Telecommunications*, 3(3):60–63, 2012.

[3] Soren Christensen and Torben Bisgaard Haagh. Design/CPN overview of CPN ML syntax. *University of Aarhus*, 3, 1996.

[4] Abdelsalam Heddaya and Abdelsalam Helal. Reliability, availability, dependability and performability: A user-centered view. *Boston, MA, USA, Tech. Rep*, 1997.

[5] Jason L. Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, and Kristofer S. J. Pister. System architecture directions for networked sensors. In *ASPLOS-IX Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, USA, November 12-15, 2000.*, pages 93–104, 2000.

[6] Bret Hull, Kyle Jamieson, and Hari Balakrishnan. Mitigating congestion in wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys 2004, Baltimore, MD, USA, November 3-5, 2004*, pages 134–147, 2004.

[7] Kurt Jensen and Lars Michael Kristensen. *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*. Springer, 2009.

[8] Kurt Jensen, Lars Michael Kristensen, and Lisa Wells. Coloured petri nets and CPN Tools for modelling and validation of concurrent systems. *STTT*, 9(3-4):213–254, 2007.

[9] Khanh Le, Thang Bui, Tho Quan, Laure Petrucci, and Etienne Andre. Congestion verification on abstracted wireless sensor networks with the WSN-PN tool. *Journal of Advances in Computer Networks*, 4(1), 2016.

[10] Khanh Le, Giang Trinh, Thang Bui, and Tho Quan. Probabilistic modelling for congestion detection on wireless sensor networks. In *Proceedings of 4th-2017 International Conference on Control, Decision and Information Technologies, CoDIT 2017, Barcelona, Spain,April 05-07, 2017*, 2017.

[11] Chuang Lin and Dan C. Marinescu. Stochastic high-level petri nets and applications. *IEEE Trans. Computers*, 37(7):815–825, 1988.

[12] Fei Liu and Ming Yang. Compositional colored petri net approach to multiscale modeling for systems biology. *International Journal of Modeling, Simulation, and Scientific Computing*, 5(04):1450017, 2014.

[13] Hui Liu, Zhijun Meng, Hua Wang, and Min Xu. Systematic random deployment for wireless sensor network in agricultural sampling-interpolation applications. In *Computer and Computing Technologies in Agriculture VI - 6th IFIP WG 5.14 International Conference, CCTA 2012, Zhangjiajie, China, October 19-21, 2012, Revised Selected Papers, Part II*, pages 53–59, 2012.

[14] Alan M. Mainwaring, David E. Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications, WSNA 2002, Atlanta, Georgia, USA, September 28, 2002*, pages 88–97, 2002.

[15] Michael K. Molloy. Discrete time stochastic petri nets. *IEEE Trans. Software Eng.*, 11(4):417–423, 1985.

[16] The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`.

[17] Fikret Sivrikaya, Thomas Geithner, Cuong Truong, Manzoor Ahmed Khan, and Sahin Albayrak. Stochastic routing in wireless sensor networks. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pages 1–5. IEEE, 2009.

[18] Giang Trinh, Khanh Le, Tam Bang, Quan Tram, Thang Bui, and Tho Quan. Modelling and congestion detection of wireless sensor networks: A concurrent-based approach using coloured petri nets. *International Journal of Applied Information Systems*, 11(7):1–9, Dec 2016.

[19] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, SimuTools 2008, Marseille, France, March 3-7, 2008*, page 60, 2008.

[20] Cristian Vasar, Octavian Prostean, Ioan Filip, Raul Robu, and Dan Popescu. Markov models for wireless sensor network reliability. In *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, pages 323–328. IEEE, 2009.

[21] Chieh-Yih Wan, Shane B. Eisenman, and Andrew T. Campbell. CODA: congestion detection and avoidance in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pages 266–279. ACM, 2003.

[22] Chonggang Wang, Bo Li, Kazem Sohraby, Mahmoud Daneshmand, and Yueming Hu. Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE Journal on Selected Areas in Communications*, 25(4):786–795, 2007.

[23] Nan Yao, Shaoping Wang, Yaoxing Shang, and Jian Shi. Reliability of wireless sensor network: Hotspot and critical challenges. In *IEEE 10th International Conference on Industrial Informatics, INDIN 2012, Beijing, China, July 25-27, 2012*, pages 1262–1266, 2012.

[24] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.