Research Article

Formal and Executable Specification of Random Waypoint Mobility Model Using Timed Coloured Petri Nets for WMN

Naeem Akhtar Khan,[1] Farooq Ahmad,[1] and Sher Afzal Khan[2]

[1] *Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan*
[2] *Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan*

Correspondence should be addressed to Sher Afzal Khan; sher.afzal@awkum.edu.pk

The wireless mesh network (WMN) is an emerging and cost-effective alternative paradigm for the next generation wireless networks in many diverse applications. In the performance evaluation of routing protocol for the WMN, it is essential that it should be evaluated under realistic conditions. The usefulness of specific mobility protocol can be determined by selection of mobility model. This paper introduces a coloured Petri nets (CP-nets) based formal model for implementation, simulation, and analysis of most widely used random waypoint (RWP) mobility model for WMNs. The formal semantics of hierarchical timed CP-nets allow us to investigate the terminating behavior of the transitions using state space analysis techniques. The proposed implementation improves the RWP mobility model by removing the "*border effect*" and resolves the "*speed decay*" problem.

1. Introduction

The wireless mesh network [1] is a communication network, which is a relatively new technology that has the potential to organize and configure them dynamically. WMNs are comprised of wireless mesh routers (WMRs) and mesh clients. Mesh routers have multiple interfaces and, with bridge functionality, enable the integration of mesh networks with other different wireless network domains. Mesh clients also have mesh networking functionality and can act as routers. WMNs have the potential to overcome the limitations of various other wireless networks such as ad mobile hoc networks (MANETs), wireless local area networks (WLANs), and wireless-fidelity (Wi-Fi). WMN is key wireless technology which has QoS mechanism to support various prominent applications. Routing is critical factor for analyzing data transfer from sender to destination. The common requirements of a reliable routing protocol are load balancing, congestion control, scalability, and reliability. WMN shares common features with ad hoc network so their routing protocol can be applied to WMNs; however mobility management for WMNs is much different than ad hoc networks and there is much room for improvement of mobility in WMNs.

Mobility in WMNs means that nodes are free to move in any direction any time with any speed arbitrarily. Therefore, the node mobility is key research paradigm in WMNs. Simulation is traditional tool for evaluating performance of networks but literature reflects that in some cases simulators generate unrealistic results which are not according to the specifications of the model. Some performance related issues of RWP have been identified in the literature. Due to so-called *border effect* [2] the model concentrates in the middle of its network environment as a spatial distribution of nodes is arbitrary. Many research results concluded this limitation of RWP mobility model for its implementation in wireless networks [3, 4]. The *speed decay* issue is another undesired behavior of RWP as reported in [5]; it shows that the average speed measured at beginning of the simulation is different from the ending speed and mean average speed is gradually decreasing with the passage of time.

Formal methods [6] have a great potential and proven to be a powerful tool for investigation of correctness of WMN routing protocols. The key factor, using formal methods in protocol engineering, is that it reduces the "time to market" and results obtained are unambiguous. Formal methods have rigorous computer based tool support for the requirements,

design, and testing of systems. Formal methods adjunct tools examine the complete state space of a model for its verification and error avoidance.

Coloured Petri nets (CP-nets) [7, 8] are formal methods suitable for specification, modeling, and verification of systems and applied for analysis of various communication protocols [9]. CP-nets are a graphically oriented modeling language suitable for expressing concurrency, on-determinism, communication, and validation of systems. CP-nets combine Petri nets [10] and the functional programming language Standard ML (SML) [11]. Petri nets provide the graphical foundation to model the concurrency, resource sharing, communication, and synchronization in the systems [12–14] whereas SML is used for modeling sequential aspects of the systems like data manipulation and designing parameter-driven models. Mobility models are used to mimic the realistic behavior of movement.

In this paper, two-dimensional $100 \times 100\,\mathrm{m}^2$ terrains are constructed for random waypoint (RWP) mobility model, a benchmark, and the most utilized model in performance analysis for wireless networks. We apply CPNs and supporting CPN tools [15], which are based on CPN ML language, to model the RWP scenario. The proposed implementation improves the RWP mobility model by removing the "border effect" and resolves the "speed decay" problem. Previously a little work, for mobility patterns by using CP-nets, has been done in the literature. The proposed model provides a foundation and it can play a vital role for evaluating the performance of routing protocols through CP-nets.

The rest of the paper is organized as follows. Section 2 describes the related work in this domain. Section 3 describes brief introduction to the traditional RWP mobility model. Section 4 presents implementation of random waypoint mobility model by using CPN tools. Section 5 describes state space analysis of model and verification results. Finally Section 6 contains conclusions and summarizes our achievement.

2. Related Work

The movement scenario of mobile hosts plays a pivot role for evaluating in performance analysis of wireless communication networks scenarios. The literature reflects that much work has not been done on mobility issues of the wireless networks by using CP-nets. In [16], the authors developed CP-nets models with respect to macromobility and micromobility scenarios. The mobile node movement between different networks is called macromobility while movement within ad hoc network is called micromobility. The CPN model abstractly describes the communication and mobility specifications and communication scenarios in the ad hoc networks. A prominent feature of the CPN model is that it reflects the simultaneous communication and the mobility functionalities in a single model. This mode is also suitable for communication and mobility test cases. The authors in [17] suggested topology approximation (TA) structure to overcome the dynamically changing topology problem. The results showed that TA mechanism can mimic the dynamic changes of network topology. Synthetic RWP

mobility model by using CPN has been presented in [18] however with some shortcomings, as the proposed model is not able to mimic the trace patterns of mobile host (MH). Khan et al. [19] presented coloured Petri nets (CP-nets) based formal model for implementation, simulation, and analysis of well-known random walk mobility model for WMN, without border effect and speed decay problems. The model is more appropriate for generating different mobility patterns; these patterns can further be used for analyzing the performance of routing protocols and mobility management in wireless communication networks. Fabulous efforts were made in [2] by removing border effect issue from RWP mobility model and spatial distribution generated by their developed model is different from standard model.

3. Theory of Random Waypoint Mobility Model

The random waypoint model was originally presented in [20] and is one of the most widely used mobility model by the researchers to simulate wireless network region. It is a benchmark synthetic mobility model due to its simplicity and availability. RWP mobility model is already implemented in NS2 and QUALNET (GLOMOSIM) simulators. In this model each mobile node is independent to other and selects random destination from direction range $[0, 2\pi]$ in the simulation area. It then moves towards new chosen destination with selected speed chosen from predefined ranges [Speed$^{\mathrm{min}}$, Speed$^{\mathrm{max}}$], where the parameter Speed$^{\mathrm{max}}$ is the maximum speed for MH. Every node selects speed and direction independent to other MH. Upon completing the epoch, the MH may pause for a certain period chosen from uniform distribution between [P^{min}, P^{max}] defined by the "pause time" factor. If the pause time P pause = 0, this enables node to continue its movement. After spending the pause time, host again decides new speed and random point in the simulation areas and travels towards it in a steady speed and complete one movement epoch. The above process is independently repeated again until the simulation accomplishes. In the RWP model, speed and pause time are the two prominent metrics that determine the mobility performance of the mobile nodes. The network remains stable, if MH velocity is low and the pause time is long, whereas, if the node moves fast and the pause time P is short, then the topology is highly dynamic. So by varying these two parameters, various mobility patterns can be obtained.

4. CPN Based Modeling for Random Waypoint Mobility Model

4.1. Formal Definition of Petri Nets. It is a cumbersome task to create intricate models, but like other modular programming languages CPN supports hierarchical module concept, in which large nets can be broken into smaller components. This section presents some basic Petri net terminology and notations which will be used in this paper.

Definition 1 (Petri nets). The Petri nets can be considered as a directed bipartite directed multigraph, containing two types

of nodes called *places* (circles) and *transitions* (rectangles) and *arcs* that connect them. It is important to note that two places or two transitions cannot be connected consecutively.

A Petri net is a four-tuple, with an initial marking M_0:

PN $= (P, T, I, O)$, where

P: a finite set of places, $\{p_1, p_2, \ldots, p_n\}$,

T: a finite set of transitions, $\{t_1, t_2, \ldots, t_s\}$,

I: an input function, $(T \times P) \rightarrow \{0, 1\}$,

O: an output function, $(T \times P) \rightarrow \{0, 1\}$.

A place p is an input place of a transition t if there exists an input arc from place to transition. Place p can be called as an output place of transition t if there exists an output arc from transition to place.

Definition 2 (coloured Petri nets). The coloured petri net is an extension of Petri net, which contains tokens, differentiated by "colour." A coloured petri net is a nine-tuple satisfying the below requirements:

CPN $= (P, \Sigma, T, A, V, C, G, E, I)$, where

(1) P is a finite set of places.

(2) Σ is a finite set of nonempty types, also called color sets. It determines the data values, operations, and functions that can be used in the net expressions.

(3) T is a finite set of transitions T such that $P \cap T = \phi$.

(4) $A \subseteq P \times T \cup T \times P$ is a set of directed arcs.

(5) V is a finite set of typed variables such that type $[v] \in \Sigma$ for all variables $v \in V$.

(6) $C : P \rightarrow \Sigma$ is a colour set function that assigns a colour set to each place. This means that each token on p must have a data value.

(7) $G : T \rightarrow \text{EXPR}_V$ is a guard function that maps each transition t in to a Boolean expression such that type$[G(t)] = $ Bool.

(8) $E : A \rightarrow \text{EXPR}_V$ is an arc expression function that maps each arc, a, into an expression of type $[E(a)] = C(p)$ multiset, where p is the place connected to the arc a.

(9) $I : P \rightarrow \text{EXPR}/0$ is an initialization function that maps each place p into a closed expression which must be of Type$[I(p)] = C(p)$ multiset.

Definition 3 (hierarchical coloured Petri nets). The following definition formally summarizes the syntax and semantics of hierarchical CPN.

A hierarchical coloured Petri net is a four-tuple CPN$_H$ = $(S, \text{SM}, \text{PS}, \text{FS})$ [21], where

(1) S is a finite set of modules. Each module is a coloured Petri net module $s = ((P^s, T^s, A^s, \Sigma^s, V^s, C^s, G^s, E^s, I^s), T_{\text{sub}}^s, P_{\text{port}}^s, PT^s)$. It is required that $(P^{s1} \cup T^{s1}) \cap (P^{s2} \cup T^{s2}) = \phi$ for all $s1, s2 \in S$ such that $s1 \neq s2$.
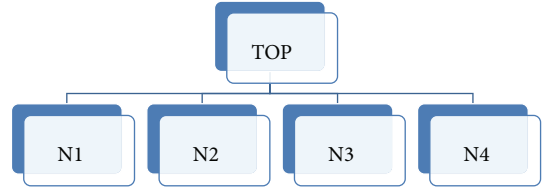

FIGURE 1: Instance hierarchy for the random waypoint mobility model.

(2) SM : $T_{\text{sub}} \rightarrow S$ is a submodule function that assigns a submodule to each substitution transition. It is required that the module hierarchy is acyclic.

(3) PS is a port-socket relation function that assigns a port-socket relation PS$(t) \subseteq P_{\text{sock}}(t) \times P_{\text{port}}^{\text{SM}(t)}$ to each substitution transition t. It is required that ST$(p) = PT(p')$, $C(p) = C(p')$, and $I(p)\langle\rangle = I(p')\langle\rangle$ for all $(p, p') \in \text{PS}(t)$ and all $t \in T_{\text{sub}}$.

(4) FS $\subseteq 2^P$ is a set of nonempty fusion sets such that $C(p) = C(p')$ and $I(p)\langle\rangle = I(p')\langle\rangle$ for all p, $p' \in fs$ and all $fs \in \text{FS}$.

For of a hierarchical coloured Petri net CPN$_H$, following concepts are defined [21].

(1) A marking M is a function that maps each compound place $[p^*]$ into a multiset of tokens $M([p^*]) \in C(p)_{\text{MS}}$, where (p, s^*) is any place instance belonging to $[p^*]$.

(2) The initial marking M_0 is defined by $M_0([p^*]) = I(p\langle\rangle$ where (p, s^*) is any place instance belonging to $[p^*]$.

(3) The variables of a transition instance t^* of a transition t are denoted by Var(t^*) and defined by Var$(t^*) = $ Var(t).

(4) A binding of a transition instance t^* of a transition $t \in T - T_{\text{sub}}$ is a function b that maps each variable $v \in \text{Var}(t^*)$ into a value $b(v) \in \text{Type}[v]$. The set of all bindings for a transition instance t^* is denoted by $B(t^*)$.

(5) A binding element is a pair (t^*, b) such that t^* is a transition instance of a transition $t \in T - T_{\text{sub}}$ and $b \in B(t^*)$. The set of all binding elements BE(t^*) for a transition instance t^* is defined by BE$(t^*) = \{(t^*, b) \mid b \in B(t^*)\}$. The set of all binding elements in CPN$_H$ is denoted by BE.

(6) A step $Y \in \text{BE}_{\text{MS}}$ is a nonempty, finite multiset of binding elements.

The CPN model of RWP mobility model is organized in a set of modules, because it is impractical to draw a large system in a flat or single net. Other reason is that often system components are used repeatedly, so it is inefficient to draw these components from scratch every time, as shown in Figure 1; same instance of RWP module is repeated as N1, N2, N3, and N4. The relationship between modules of
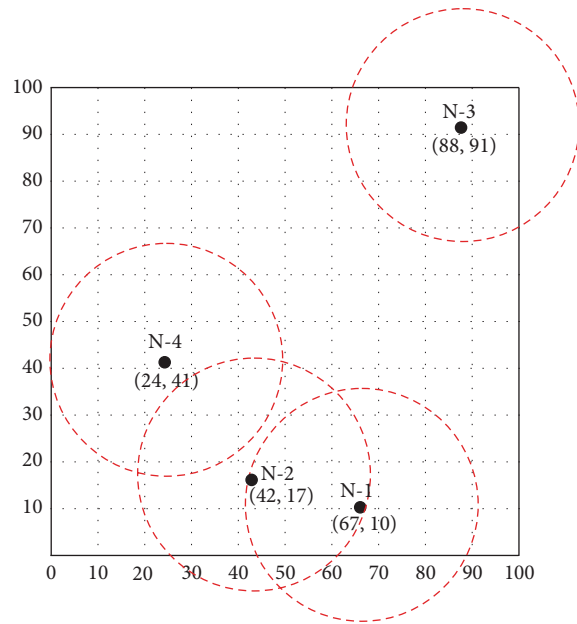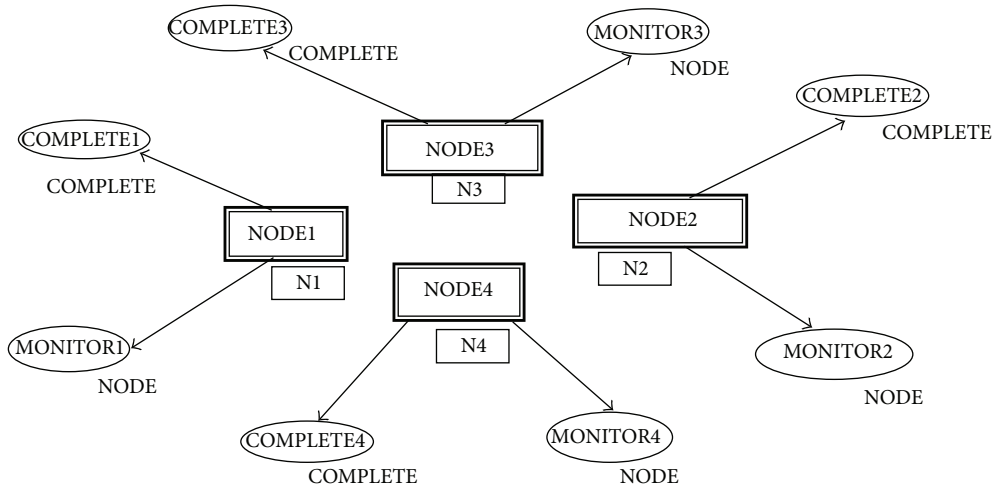
FIGURE 2: RWP based nodes topology.

FIGURE 3: Scenarios page, top level page.

our proposed model is depicted in Figure 1; the modules names are written inside the nodes. The TOP module has no incoming arcs, so it is called the prime module; next N1 module has four instances.

4.2. Environment Scenario. The random waypoint CPN model consists of 5 pages. Four nodes are spread randomly in a $100 \times 100 \, \text{m}^2$ communication area. Node N1 is randomly placed at (67, 10), N2 at (42, 17), N3 at (88, 91), and N4 at (24, 41) as shown in Figure 2. Model is as abstract view; however nodes can be enhanced. Approximate transmission range of 802.11n based MH transmitter is 250 m which is shown by circle. Time clock is considered in minutes.

The hierarchical CPN model is depicted in Figure 3, in top level page each MH is reflected by 2 places *"COMPLETE"* and *"MONITOR"* and one substitution transition "NODE" which has its associated subpage.

4.3. The Color Set. The CPN ML language provides a predefined set of basic types inherited from SML functional programming language for declaring operations, variables, functions, constants, expressions, bindings, and colour set (types). The proposed RWP model has a set of declarations and introduces a number of colour sets. The colour sets are used to declare new colour sets like products and records; some important colour sets are used to construct RWP model which is reflected in Table 1 to elaborate their functionality.

4.4. The Functions. It is convenient to write a complex expression as a function, similar to concept of methods and

TABLE 1: The color set.

| Color Set	Definition
colset INT = int	Integer color set
varsp, bb, c, w, f, dir, ax: INT;	Integer type variables
colset ID = int	Node identity
var id, hx, hy, cx, cy, dx, dy: ID	Nodes variables
colset COOR = product ID ∗ ID	Product type coordinates (x-axis, y-axis)
colset STEP = int	Movement steps
var steps: STEP	Steps type variable
colset STATUS = with HOME | MOVING | FINISH	Three states if MH, home, moving, or finish (reached to destination)
colset NODE = product ID ∗ STEP ∗ COOR ∗ COOR ∗ COOR ∗ STATUS timed	Timed node color set 6-tuple consists of node ID, step number, home, current, destination, and current status
var n: NODE	Node type variable
colset COMPLETE = product ID ∗ STEP ∗ COOR ∗ COOR ∗ COOR ∗ STATUS ∗ INT timed	Product type color set same as NODE but it includes 1 more tuple after STATUS as INT, which increased by 1 on every successful epoch
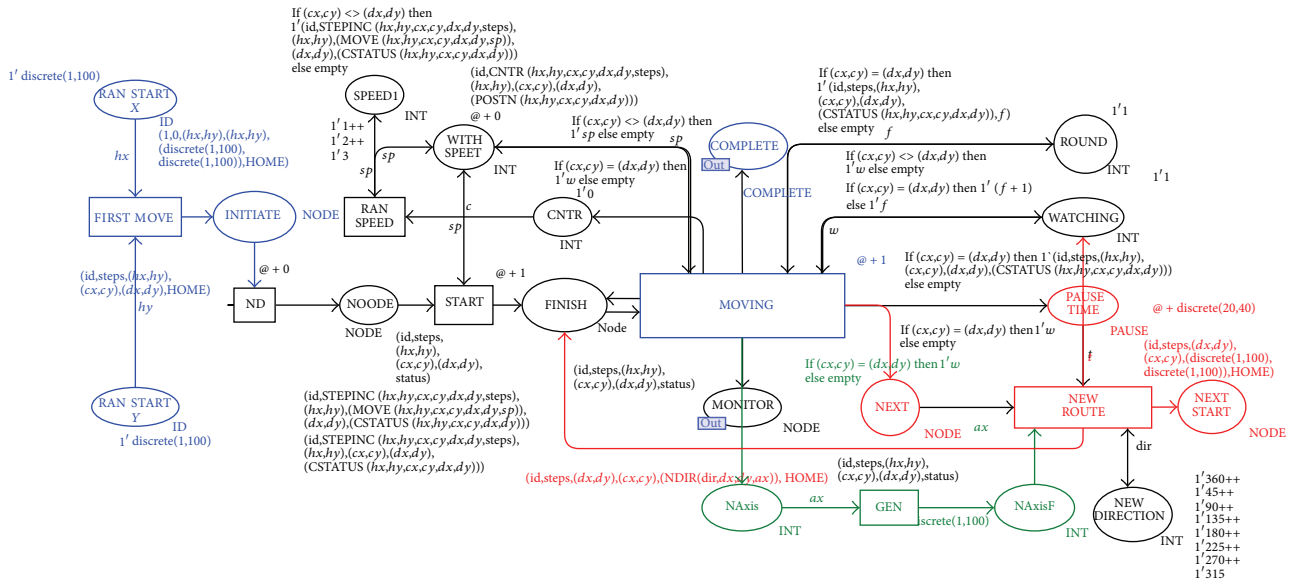
FIGURE 4: The CPN model of RWP mobility model.

procedures from conventional programming languages. The keyword fun is used to declare a function in CPN ML. Table 2 shows definitions of some important functions used to capture the desired behavior from proposed model.

In CPN model, states of a system which are represented by places and operations or events are modeled by transitions as shown in Figure 4. In the initial marking there is one token in the places *RANSTARTX*, *RAN STARTY*, *CNTR*, *ROUND*, and *WATCHING*, three tokens in place *SPEED1*, and eight tokens in place *NEW DIRECTION*, respectively. The initialization step of proposed model is responsible for initial distribution of tokens in initial marking. However it is possible for a user to manually specify initial marking. It is further contended; in our specific case we placed different number of tokens on different places, as per requirement of our model. For example, eight tokens of integer type in place *NEW DIRECTION* specify 8 angles from 0–360 degrees, that is, 45°, 90°, 135°, 180°, 22°, 270°, 315°, and 360°.

On completion every epoch MH selects new direction for next destination. On the other hand place SPEED1 contains three tokens of integer colour with values 1, 2, and 3 and a user can add more speed tokens or enhance the value. These tokens represent speed of the epoch and model randomly chooses one speed token for every destination. The single token in place *RANSTARTX* and *RANSTARTY* is used to identify home coordinates of MH. In the same way the places, *CNTR*, *ROUND*, and *WATCHING*, hold single token for speed counter, counting number of rounds, and monitoring current status of MH, respectively.

4.5. Traveling Sequence of Mobile Node with respect to RWP. As simulation starts, based on RWP mobility model specification place *RAN START X* and *RAN START Y* produce randomly x-axis and y-axis coordinates from terrain range, that is, $100 \times 100\,\text{m}^2$, which enables transition

Table 2: The function.

Function	Definition
fun CNTR	This function is used to determine the counter of steps, if home coordinates are equal to current coordinates then steps will remain same; otherwise steps will increase by 1 on each move
Fun POSTN	This is a flag function, which depicts 3 states of MN. 3 flags are "HOME," "MOVING," and "FINISH." When MN is at first coordinate the HOME tag is shown in traveling status, when moving towards destination then "MOVING" and at last when reached its destination then "FINISH" flag is reflected in status like $1'(1, 0, (67, 10), (67, 10), (43, 66), HOME) @ 0$
fun MOVE	This is an important function, which is responsible for MH movement as per requirement of requisite speed and direction
fun STEPINC	This function is used to increment step counter to be 1 on each move
fun CSTATUS	This function is used to check whether MH has reached to the destination or not and updates statues accordingly
Fun NDIR	This function decides the direction of MH for destination

Table 3: Movement pattern of 4 NODES after 1200 steps simulation.

NODE 1, completed 7 epoch in 490 m

$1'(1, 28, (67, 10), (11, 21), (11, 21), FINISH, 1) @ 29{+}{+}{+}$
$1'(1, 69, (11, 21), (92, 8), (92, 8), FINISH, 2) @ 99{+}{+}{+}$
$1'(1, 95, (92, 8), (16, 16), (16, 16), FINISH, 3) @ 146{+}{+}{+}$
$1'(1, 179, (16, 16), (100, 100), (100, 100), FINISH, 4) @ 256{+}{+}{+}$
$1'(1, 183, (100, 100), (92, 100), (92, 100), FINISH, 5) @ 289{+}{+}{+}$
$1'(1, 280, (92, 100), (97, 3), (97, 3), FINISH, 6) @ 424{+}{+}{+}$
$1'(1, 307, (97, 3), (20, 3), (20, 3), FINISH, 7) @ 490$

NODE 2, completed 11 epoch in 530 m

$1'(2, 57, (42, 17), (34, 74), (34, 74), FINISH, 1) @ 58{+}{+}{+}$
$1'(2, 67, (34, 74), (34, 64), (34, 64), FINISH, 2) @ 106{+}{+}{+}$
$1'(2, 73, (34, 64), (16, 64), (16, 64), FINISH, 3) @ 136{+}{+}{+}$
$1'(2, 125, (16, 64), (68, 68), (68, 68), FINISH, 4) @ 211{+}{+}{+}$
$1'(2, 133, (68, 68), (60, 68), (60, 68), FINISH, 5) @ 242{+}{+}{+}$
$1'(2, 143, (60, 68), (32, 68), (32, 68), FINISH, 6) @ 281{+}{+}{+}$
$1'(2, 181, (32, 68), (18, 18), (18, 18), FINISH, 7) @ 347{+}{+}{+}$
$1'(2, 194, (18, 18), (44, 18), (44, 18), FINISH, 8) @ 384{+}{+}{+}$
$1'(2, 196, (44, 18), (38, 18), (38, 18), FINISH, 9) @ 423{+}{+}{+}$
$1'(2, 224, (38, 18), (73, 73), (73, 73), FINISH, 10) @ 487{+}{+}{+}$
$1'(2, 230, (73, 73), (67, 73), (67, 73), FINISH, 11) @ 530$

FIRST MOVE. After firing *FIRST MOVE* transition, one token moves to place *INITIATE* with parameters 1, 0, (60, 47), (60, 47), (6, 20), HOME) @ 0, in this tuple 1 is *Node ID*, 0 is *Step number*, (60, 47) is home coordinates, (60, 47) reflects current coordinates, (6, 2) shows destination, *HOME* means initial state, and @ 0 reflects time of model. By firing transition *ND* token moves to place *NODE*, here *CNTR* function is implemented for increasing steps. Transition *RAN SPEED* generates speed for node movement and, after its occurrence, the transition *START* enables, which takes 10 parameters as input and sends token to place *FINISH* after calculation next coordinates, adds next step and delay of time @ 1. In *START* transition three functions are used, first to calculate next step function *STEPINC* is implemented, *MOVE* function is used to move next step in desired direction, and finally *CSTATUS* is used to flag the status of current scenario. After firing transition *START,* token moves to *FINISH* place and enables *MOVING* transition, which takes 12 input parameters and decides the next move of MH; if current and destination tuples are the same, its mean node has reached its destination. Otherwise it increases step and next coordinates in the nodes payload and send back to *FINISH* place to repeat the process until node reaches its destination.

When one epoch is completed, token moves to the place *NEXT*, and pause time is added by uniform random distribution (20, 40) function. *NEW ROUTE* transition takes 12 inputs and generates new epoch for *MH* and sends mobility information token to the place *FINISH* after adding pause time. *FINISH* place repeats the process until simulation ends. Figure 5 illustrates the top page after 1200 steps with time stamp 560. Top page is showing 4 nodes, that is, N1, N2, N3,

and N4, each with 1 substitution transition and two places *COMPLETE* and *MONITOR*.

Place *COMPLETE1* as an output of component *Node1* showing seven tokens, that is, $(1, 28, (67, 10), (11, 21), (11, 21), FINI + SH, 1) @ 29, (1, 69, (11, 21), (92, 8), (92, 8), FINISH, 2) @ 99 (1, 95, (92, 8), (16, 16), (16, 16), FINISH, 3) @ 146 (1, 179, (16, 16), (100, 100), (100, 100), FINISH, 4) @ 256 (1, 183, (100, 100), (92, 100), (92, 100), FINISH, 5) @ 289 (1, 280, (92, 100), (97, 3), (97, 3), FINISH, 6) @ 424 (1, 307, (97, 3), (20, 3), (20, 3), FINISH, 7) @ 490$. These mobility payloads reflect that node *ID* 1 completed 7 epochs, 1st round with time stamp 29, after random pause time, started 2nd round and finished at time 99, 3rd epoch is completed at time 146, 4th epoch is finished @ 256, 5th @ 289, 6th @ 424 and 7th epoch is completed at time 490, respectively. In the same manner node 2 moved in the terrain as depicted in Table 3.

The place *MONITOR1* has 348 tokens, showing traveling pattern from *HOME to DESTINATION* of N1. The places *MONITOR2*, *MONITOR3*, and *MONITOR4* have 240, 255 and 227 tokens, respectively, reflecting movement trace patterns of MHs N2, N3, and N4.

5. Simulation Analysis and Discussion

In order to simulate and to verify the proposed RWP mobility model we use CPN tool; through state space analysis we can analyze the trace pattern generated by our model.

Table 4, showing the statistics, reveals that O-graph has 30,895 nodes and 39,514 arcs; there exists strongly connected graphs (SSG), and so model has infinite occurrences
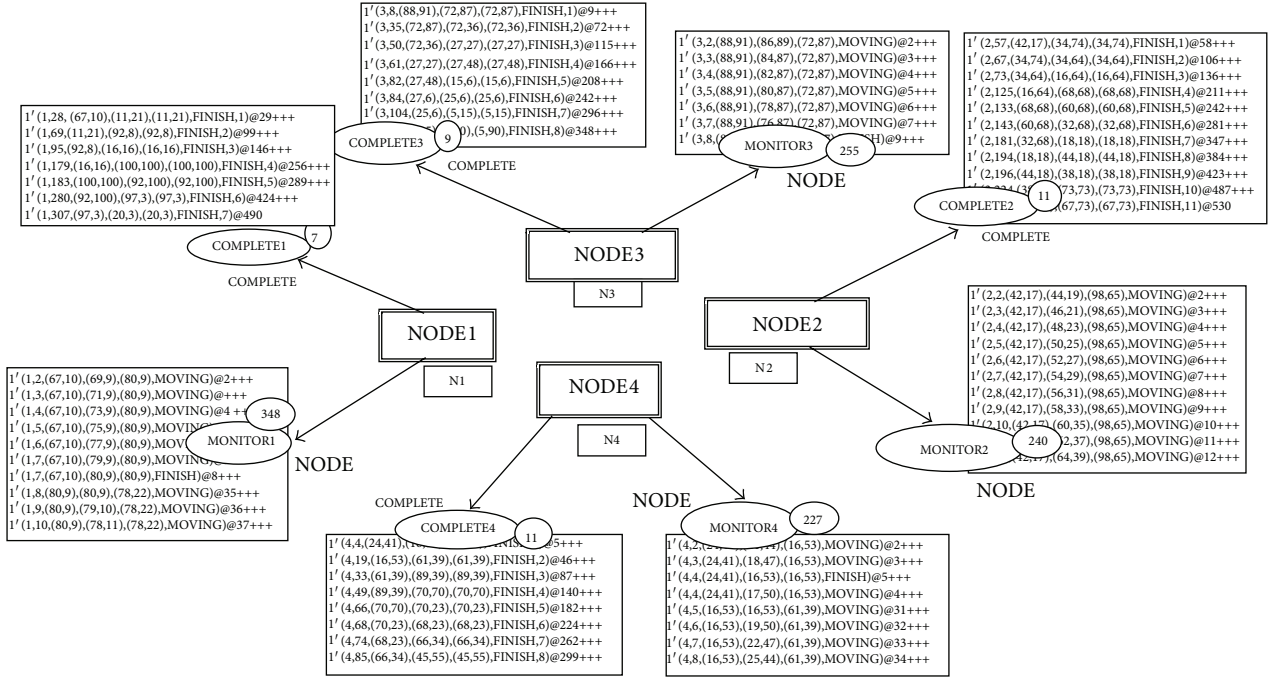
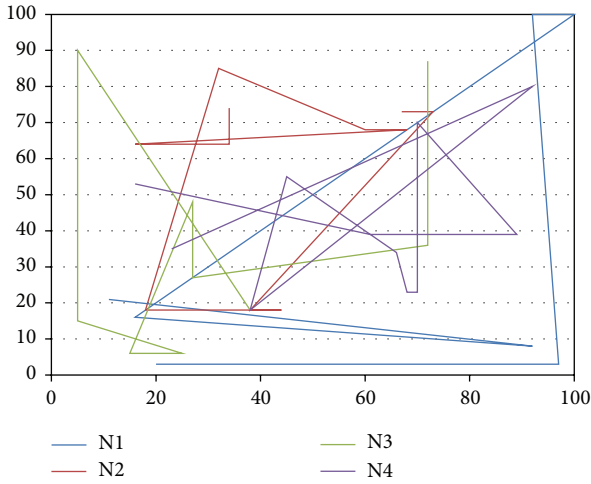FIGURE 5: Traveling pattern of MH based on RWP mobility model.

FIGURE 6: *HOME* To *DESTINATION* moving pattern of MH based on Table 3.

TABLE 4: Statistics for o-graph.

State Space		
Nodes: 30895		
Arcs: 39514		
Secs: 300		
Status: Partial		
Scc Graph		
Nodes: 30895		
Arcs: 39514		
Secs: 4		
Boundedness Properties		
Best Integer Bounds	Upper	Lower
N1$'$CNTR 1	1	0
N1$'$FINISH 1	1	0
N1$'$INITIATE 1	1	0
N1$'$NAxis 1	0	0
N1$'$NAxisF 1	0	0
N1$'$NEW_DIRECTION 1	8	8
N1$'$SPEED1 1	3	3
N1$'$WATCHING 1	1	1
N1$'$WITH_SPEET 1	1	0
Dead Markings		
26780 [9999, 9998, 9997, 9996, 9995, . . .]		
Live Transition Instances		
None		

sequences. MH can move in communication range without any restriction.

HOME to *DESTINATION* paths of four mobile nodes have been reflected in Figure 6; the transmission range and the traveling path of every node as well as its neighbors can be seen, where node can communicate.

Liveness property specifies that every transition can be enabled from each marking by firing an arbitrary number of transitions. The RWP represents unlimited navigation within the system. Figure 7 gives some results from standard queries. As an example we can find path from M^1 to M^{26780} by *NodesInPath* (1, 26780) query.

Upper and lower bound properties are very prominent for analyzing; the upper and the lower bound tell the maximum and minimum number of tokens which a specific place can contain. This analysis is part of simulation, so it can

NodesInPath(1,26780)   Val it = [1,3,43,405,3516,26780]: node list

Reachable(1,26780)   Val it = true : bool

Reachable(26780,1)   Val it = false : bool

SccReachable(1,26780)   Val it = true : bool

SccReachable(26780,1)   Val it = false : bool

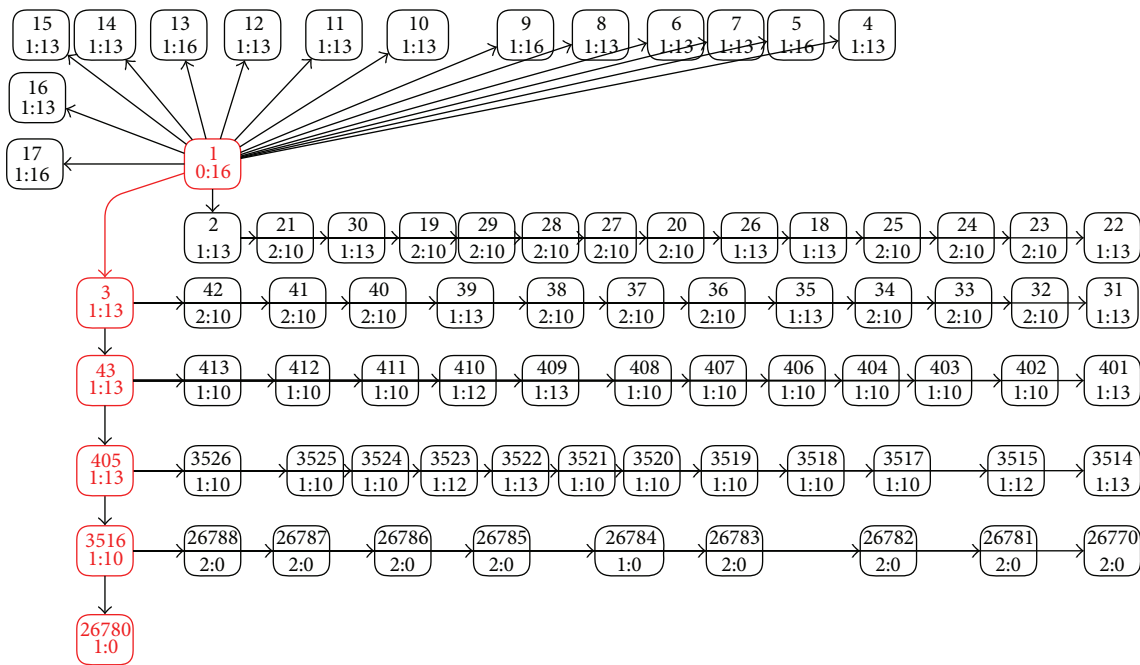FIGURE 7: Occurrence sequence marking M^1 to M^{26780}.

FIGURE 8: State space analysis, O^{26780}.

be different in next simulation. The places *CNTR*, *FINISH*, *INITIATE*, *NOODE*, *RAN_START_X*, *RAN_START_Y*, and *WITH SPEET* can contain either 0 or 1 token. The place *NEW_DIRECTION* contains 8 tokens at each marking. It is a direction place and system can select one random direction for each epoch. Moreover, the place *SPEED1* contains three tokens for every marking. It is a speed place and system can select one random speed for each epoch. The place *WATCHING* contains 1 token in every state.

Traverse path form initial marking M^1 to M^{26780} is investigated in Figure 8, and the shortest path is constructed. It is shown in Figure 8 that there are minimum four markings which must occur for constructing occurrence sequence from M^1 to M^{26780}. It is shown that 26780 (9999, 9998, 9997, 9996, 9995, . . .) numbers are dead markings. M^{26780} tells us that there is 1 incoming arc and 0 outgoing arcs. It can be interpreted as a final marking for RWP mobility

model and it is not a dead lock state. The deadlock state can be explained that all instructions have been executed and no lock exists on the system. However, occurrence of a deadlock state can be interpreted that some instructions have not been fully executed; its mean system is partially correct.

Figure 9 shows speed verses number of steps of two mobile nodes. It reflects that speed is constant and not gradually decreasing during a trip. N1 completed first itinerary in 27 steps with speed 1 while speed from first to last step remained constant. Thereafter, from step number 28 MH started 2nd epoch with speed 2 and reached its destination on step number 36. N2 started first route from step number 1 and completed at step number 16 with speed 3. After that N2 started second trip with speed 2 and completed it at step number 24. Same statistics can be seen for remaining steps in Figure 9 for both the nodes.
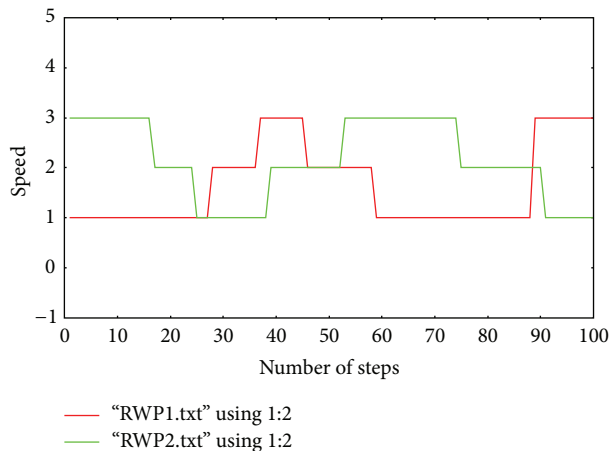
FIGURE 9: Speed of 2 mobile nodes without speed decay problem.

6. Conclusion

At the time of writing not much research work about CPNs based traditional RWP mobility model has been reflected by the literature. Most of existing mobility research is based on NS-2 simulator and we envisage that formal modeling based work, presented in this paper, will open new paradigm. The work presented in this paper is based on our efforts in developing an accurate and executable RWP mobility model without *border effect* and *speed decay* problems. Further, extensive simulations were conducted for the evaluation of proposed model. The model is more appropriate for generating different mobility patterns, and these patterns can be further implemented in analyzing the performance of routing protocols in wireless communication networks. There are several directions for future work; a more compact mobility model with tokens representing MHs can be extended. It can be easily modified and extended to capture the complex scenarios.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.

[2] G. Resta and P. Santi, "An analysis of the node spatial distribution of the random waypoint mobility model for ad hocnetworks," in *Proceedings of the 2nd International Workshop on Principles of Mobile Computing (POMC '02)*, pp. 44–50, October 2000.

[3] D. M. Blough, G. Resta, and P. Santi, "A statistical analysis of the long-run node spatial distribution in mobile ad hoc networks," in *Proceedings of the ACM MSWiM Conference*, Atlanta, Ga, USA, September 2002.

[4] D. Mitsche, G. Resta, and P. Santi, "The random waypoint mobility model with uniform node spatial distribution," *Wireless Networks*, vol. 20, no. 5, pp. 1053–1066, 2013.

[5] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 2, pp. 1312–1321, San Francisco, Calif, USA, March-April 2003.

[6] E. M. Clarke, J. M. Wing, R. Alur et al., "Formal methods: state of the art and future directions," *ACM Computing Surveys*, vol. 28, no. 4, pp. 626–643, 1996.

[7] K. Jensen, "Basic concepts, analysis methods and practical use," in *Coloured Petri Nets*, vol. 1–3 of *Monographs in Theoretical Computer Science*, Springer, Berlin, Germany, 1997.

[8] L. M. Kristensen, S. Christensen, and K. Jensen, "The practitioner's guide to coloured petri nets," *International Journal on Software Tools for Technology Transfer*, vol. 2, no. 2, pp. 98–132, 1998.

[9] J. Billington, M. Diaz, and G. Rozenberg, Eds., *Application of Petri Nets to Communication Networks: Advances in Petri Nets*, vol. 1605 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 1999.

[10] W. Reisig and G. Rozenberg, Eds., *Lectures on Petri Nets: Advances in Petri Nets. Volume I: Basic Models*, vol. 1491 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 1998.

[11] J. D. Ullman, *Elements of ML Programming*, Prentice-Hall, 1998.

[12] F. Ahmad and S. A. Khan, "Specification and verification of safety properties along a crossing region in a railway network control," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 5162–5170, 2013.

[13] F. Ahmad and S. A. Khan, "Module-based architecture for a periodic job-shop scheduling problem," *Computers and Mathematics with Applications*, vol. 64, no. 1, pp. 1–10, 2012.

[14] A. Sadiq, F. Ahmad, S. A. Khan, J. C. Valverde, T. Naz, and M. W. Anwar, "Modeling and analysis of departure routine in air traffic control based on Petri nets," *Neural Computing & Applications*, 2014.

[15] CPN Tools, http://www.daimi.au.dk/CPNtools.

[16] L. M. Kristensen, J. B. Jørgensen, and K. Jensen, *Application of Coloured Petri Nets in System Development*, Lectures on Concurrency and Petri Nets, 2003.

[17] C. Xiong, T. Murata, and J. Tsai, "Modeling and simulation of routing protocols for mobile ad hoc networks using colored petri nets," in *Proceedings of the Conference on Application and Theory of Petri Nets: Formal Methods in Software Engineering and Defence Systems*, vol. 12 of *Research and Practice in Information Technology*, pp. 145–153, 2002.

[18] S. Agrahari and S. Chinara, "Simulation of random waypoint mobility model using colored Petri nets," in *Proceedings of the 2nd International Conference on Computer Science, Engineering and Applications (CCSEA '12)*, pp. 65–75, 2012.

[19] N. A. Khan, F. Ahmad, and W. A. Baig, "Formal model of random walk mobility model for WMN using coloured petri nets," in *Proceedings of the 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST '14)*, pp. 436–444, 2014.

[20] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imelinsky and H. Korth, Eds., pp. 153–181, Kluwer Academic Publishers, Boston, Mass, USA, 1996.

[21] K. Jensen and L. Kristensen, *Coloured Petri Nets Modeling and Validation of Concurrent Systems*, Springer, New York, NY, USA, 2009.