DOI: 10.21276/sjet.2016.4.6.2

Scholars Journal of Engineering and Technology (SJET) **ISSN 2321-435X (Online)**
Sch. J. Eng. Tech., 2016; 4(6):271-276 **ISSN 2347-9523 (Print)**

(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

Research Article

A Method of Behavioral Analysis of OpenFlow Protocol

Isaam Saad, Tkachova O
Telecommunication systems department, 61166, Kharkiv National University of Radioelectronics, Ukraine

***Corresponding author**
Isaam Saad
Email: saadissam@ymail.com

Abstract: The paper is devoted to constructing a new method of analysis functional and non-functional requirements of OpenFlow protocol realization. The modified reachability tree for compliance and behavior analysis of model of OpenFlow protocol is proposed in the paper. The proposed method bases on modeling approach and give ability analyze the protocol functionality on earlier stages of realization. The E-net math tool and ordinary reachability tree method are taken into account. The existing of management transition in E-net model gives ability to eliminate a symbol ω, the main leak of ordinary methods. Thus proposed method shows numerical activity of transition and help to contract branches that lead to deadlocks and loops.
Keywords: reachability tree, OpenFlow protocol, E-net, behavior analysis, functional requirement, activity of transition.

INTRODUCTION

The main feature of OpenFlow protocol is a big amount of different programming realization. The programming realization of OpenFlow protocol, their functioning and non-functioning characteristics are significantly different between each other. This is the reason of mistakes during network functionality. Therefore, the task of compatibility checking of programming realizations of OpenFlow protocol has actuality in the implementation and maintenance process. As a rule, the compatibility analysis or behavior analysis is performed by mathematical methods of on different stages of protocol implementation. In addition, the approach effectively used to search contradictions and inconsistencies in protocol requirements.

The OpenFlow protocol involves interaction of a large number of processes that have both synchronous and asynchronous nature. Behavioral analysis of such processes types in communication protocols are performed by Petri net and its derivatives. There are number of research works that apply the Petri net approach to the behavior analysis of protocols on transport and application layers of OSI models [1-3]. The high level of visibility and height-developed methodology are the main advantages of Petri net approaches. However, this approaches cannot be apply to OpenStack protocol correctly because Petri nets do not allow establishing and controlling the relationship between processes states.

E-nets as math tool is widely used for modeling systems with a plurality of parallel processes that also have both synchronous and asynchronous nature. E-nets tool application for OpenFlow protocol behavior modeling and subsequent analysis of algorithmic properties E-net model provides the good opportunity checking the correctness of proposed programing realization of OpenFlow [4].The management transition is used for correct interpretation of process behavior in OpenFlow protocol. Compliance results of the protocol realization to primary can also be performed in such way.

Methods based on reachability tree buildings for existing OpenFlow protocol model [5], matrix methods [2] and methods of reduction [3] are widely used as means for models analysis [7]. However, usage of this method for the analysis of E-net models properties faces with number of difficulties.

The reachability tree gives ability to find all the feasible set of labeling model protocol and check the activity of each transition in the model of protocol. Research the labels changing in model and future in reachability tree gives opportunity to analyze sequence order when the transitions are active. The appearance of symbol ω in process of reachability tree building [1-3] is a significant disadvantage for analysis of such properties as boundaries, limitedness and safety.

The introduction of a finite set of critical positions that characterized by a set of attributes is one of the ways to eliminate this disadvantage. For example, the operation of counters should presence of one incoming and outgoing arc for each place.

Thus, the performing of behavioral analysis of OpenFlow protocol is the most appropriate means of E-net models. Analysis of algorithmic properties of E-net model OpenFlow protocol will provide the full specter of functional and non-functional characteristics. The article devoted to improve the efficiency of the analysis of the properties of the models and the development of a modified method for constructing the reachability tree.

A METHOD OF REACHABILITY TREE CONSTRUCTION

The existing of management transition MX, MY with management elements $r(x_1, x_2, ...x_n)$, where $x_1, x_2, ...x_n$ are the attributes [4, 7]. Assume that counter that can be set at any transition is one of this attribute. In this way, counter give ability to replace symbol ω for specific numerical value. This replacement will be possible do to set counter attribute in the management transitions. Thus, the solution gives ability to check compliance of non-functional requirements, which contain numerical characteristics. Existing only one incoming and outgoing arc that belong one position of E-net model restricts reproduction of labels. It avoids unauthorized changes counters to manage the transition by increasing the number of labels. Algorithm of modified algorithm of reachability tree building is depicted on Figure 1.

The modified algorithm of reachability tree takes into account features that mention above. The proposed method of reachability tree construction includes several additional steps in comparison with ordinary reachability tree algorithm and consists from the following sequence of actions:

1. To identify the initial labeling M_0 of the E-net model. Where M_0 is initial model labeling.
2. To determine the potentially reachable places of the model.
3. To determine the potentially active transitions. The first potential active transition is transition that follows the initial labeled places or root place and has empty post-places.

3.1 To define a set of active transitions and a set of reachable places. Formed the order "pre-place – transition – post-place" while reachable places exist.

In this case, each reachable place can be terminal, duplicate or boundary.

3.2 To define the places type. In the case when the reachable place x and y have defined and if place x was obtain before and place y was obtained. The next main cases should be observed:

- if the label of place x (M_x) and the label of place y (M_y) have the equaling labeling, than place y is duplicating the place x;
- if the label of place x (M_x) is not equal the label of place y (M_y): $M_x \neq M_y$ and active transitions that follow y do not exist, the place y is a terminal and has terminal labeling M_y;
- if the label of place x (M_x) is not equal the label of place y (M_y): $M_x \neq M_y$ and active transitions that follow y do not exist (the post-place can be reachable), the place y is a boundary and if pre-place x is matched as internal.

3.3 To determinate of the set of transitions involved in tree reachability.

Determination of compliance of place x or label (M_x) to specific E-network position p_x, where p_x the position definition and place y (in the case when y is post-place of x to specific E-network position p_y. If pre-place and post-place have defined, the meaning of transition that change the label M_x to M_y in the model is established.

4. To perform recursively the steps 1-3 while active transition are existing. Formation of the reachability tree ends when all terminal, duplicate or boundary vertices are installed (no active transitions).

5. To establish set of active positions $\{M_f'(p_i)\}$. Where $\{M_f'(p_i)\}$ defines the whole amount of places that involved in the process of moving the labels from the root to a particular terminal node of the set of finite states or finite places (M_f).

5.1 The number of active transitions $\{M_f'(h_{i-1})\}$ for certain labled place and certain route (reachability tree branch) through all places during the route that take part in building of reachability tree.

- if place i exist and belong the path from root node to the final state i and the next rules $M_i \to M_f$, $M_i < \gamma(M_f, h_{f-1})$ $(M_i)_\omega < \gamma(M_f, h_{f-1})_\omega$ $(M_i)_\omega = \omega$ are actual, then symbol ω can be convert to $(M_i)_\omega = (0,0,...,\underset{i}{\omega},...,0)$;

- if the place with label M_f exist, where $(M_f)_\omega = \omega$ and $(M_z)_\omega = \omega$, the value ω takes the value of management transition (MX or MY).

5.2 The first appearance of character ω is determined. If the management translation contained the counter, each time when starting the transition counter value is incremented.

6. Determination a labeled vector for each branches of reachability tree.

The labeled vector is an ordered set of all reachable places, which are involved in the formation of reachability tree.

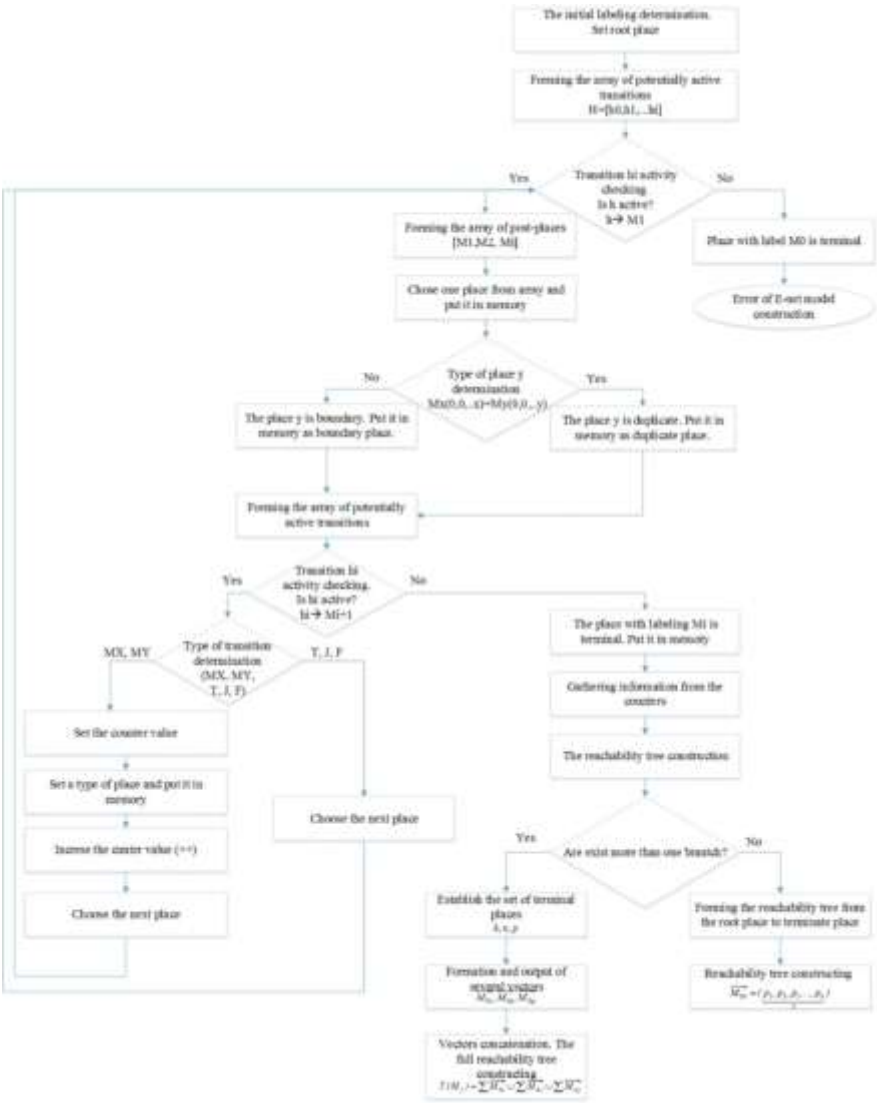

Fig-1: Flow- chart of reachability tree algorithm

Definition: Let's the initial labeling of root of reachability tree is $M_0 = (\underbrace{1,0,0,0,...,0}_{k})$, where k is amount of potentially reachable places. Label the position " p_1 " determines the position number of the model. Let the following

statements $M_0 = (\underbrace{p_1,0,0,0,...,0}_{k})$, $M_1 = (\underbrace{0,p_1,0,0,...,0}_{k})$ and $\overrightarrow{M_2} = (\underbrace{0,0,p_1,...,0}_{k})$ are true. In this case, the vector markings can be given as $\overrightarrow{M_{0k}} = (\underbrace{p_1,p_2,p_i,...,p_k}_{k})$

7. To determine the resulting labeled vector for certain finite labeling M_k. The resulting labeled vector may be defined as follows:

$$\overrightarrow{\sum M_{0k}} = (\underbrace{\sum_1^n p_1, \sum_1^m p_2, \sum_1^l p_i, ,..., \sum_1^j p_k,}_{k}) = (\underbrace{p^n{}_1, p^m{}_2, p^l{}_i, ..., p^j{}_k}_{k}). \quad (4)$$

8. To perform recursively the steps 1-7 while reachability tree branches are existing and create a plurality of reachability tree:

$$T(M_f) = \sum \overrightarrow{M_{0k}} \cup \sum \overrightarrow{M_{0n}} \cup \sum \overrightarrow{M_{0p}} =$$

$$= (\underbrace{\sum_1^n p_1, \sum_1^m p_2, \sum_1^l p_i, ,..., \sum_1^j p_k}_{k}) \cup (\underbrace{\sum_1^n p_1, \sum_1^m p_2, \sum_1^l p_i, ,..., \sum_1^j p_n}_{n}) \cup (\underbrace{\sum_1^n p_1, \sum_1^m p_2, \sum_1^l p_i, ,..., \sum_1^j p_p}_{p}) =$$

$$= (p_1^n p_2^m p_i^l ... p_k^j) \cup (p_1^h p_2^m p_i^l ... p_n^j) \cup (p_1^n p_2^m p_i^l ... p_p^j). \quad (5)$$

Restriction: The reachability tree can be building if and only if the amount of places in E-net model is finite amount.

EXPERIMENT AND RESULTS: AN APPROBATION OF THE PROPOSED METHOD

An analysis of OpenFlow switch behavior proposes in this section. We are trying to check the correctness of the switch behavior during new message formation and forwarding to the controller. The fragment of E-net model for messages exchange and establishing connection on OpenFlow switch are depicted on Figure 2.

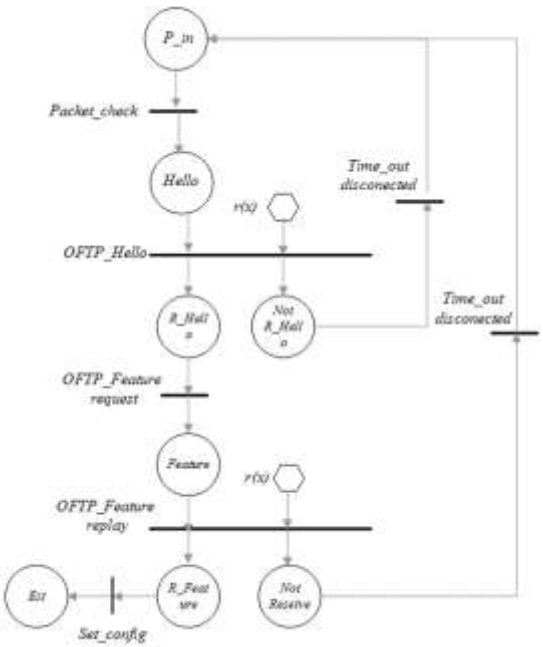

Fig-2: Model of message exchange on the OpenFlow switch

The place P_in in E-net model that depicted on Figure 2 represents the new request for establish connection. The request can arrive to any port of OpenFlow switch. The place $Hello$ represents the acknowledgment and invitation message formation on OpenFlow switch. The place R_Hello represents the received answer from controller. The place $Feature$ the formation of an information message about the switch properties. The place $R\_Feature$ represents the necessary features calibration (capacity, speed, etc.). The place Est presents the condition «connection established». Negative condition represented by next places: Not_R_Hello - invitation message has not confirmed; $Not\_R\_Feature$ - the set of proposed properties has not confirmed.

The reachability tree for E-net model that represent the OpenFlow switch behavior during process of message initialization is depicted on Figure 3.

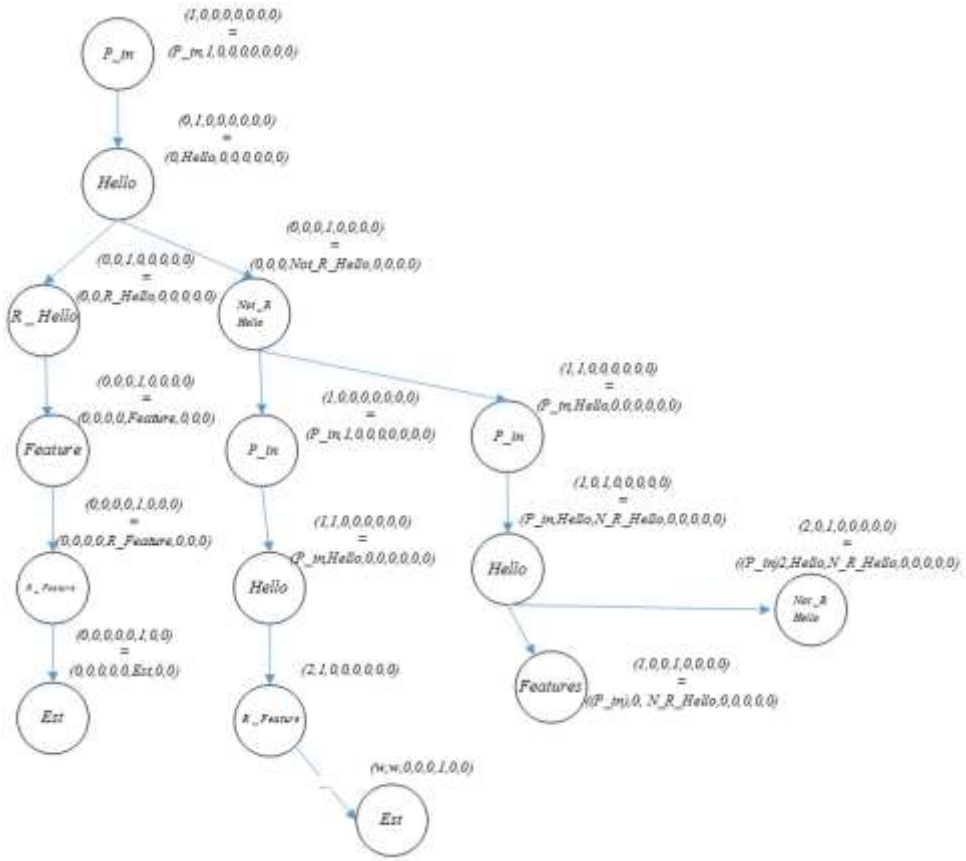

Fig-3: Reachability tree of E-net model of formation messages on OpenFlow switch

The next sequences of states are form of whole branches of tree reachability that belong to E-net model. The sequence of places is the result of performing steps 1-7 of proposed above methods:

$$
\begin{aligned}
&T(M_f) = \sum \overrightarrow{M_{0Est}} \cup \sum \overrightarrow{M_{0(N\_R\_Hello)}} \cup \sum \overrightarrow{M_{0(N\_R\_Fearute)}} = \\
&= (P\_in, Hello, R\_Hello, Feature, R\_Feature, Est) \cup \\
&\cup ((P\_in, Hello)^2, R\_Hello, Feature, R\_Feature, Est) \cup \\
&\cup ((P\_in, Hello)^3, (R\_Hello, Feature, N\_R\_Feature)^2, Est) \cup \\
&\cup (P\_in, Hello, N\_R\_Hello)^3 \cup (P\_in, Hello, R\_Hello, Feature, N\_R\_Feature)^3
\end{aligned}
\tag{6}
$$

The reachability tree branches that include the sequence change of placing to terminal states achievement is included to the statement (6).

The next four sequences of places were formed as a result of achievement the label $M(0,0,0,0,0,1,0,0)$. Where the $M(0,0,0,0,0,1,0,0)$ introduces the successful new message forming and connection establishing:

$(P\_in, Hello, R\_Hello, Feature, R\_Feature, Est)$,

$((P\_in, Hello)^2, R\_Hello, Feature, R\_Feature, Est)$,

$((P\_in, Hello)^3, (R\_Hello, Feature, N\_R\_Feature)^2, Est)$.

Should be noted that the appearance of degree index in sequences indicates that the model don't support the safety and limitedness.

The sequences $(P\_in, Hello, N\_R\_Hello)^3$ and $(P\_in, Hello, R\_Hello, Feature, N\_R\_Feature)^3$ consist the terminal nodes ($N\_R\_Feature$ and N_R_Hello) that do not lead to the successful result.

CONCLUSION

The compliance analysis of OpenFlow protocol functional and non-functional requirements allows for increased efficiency both different stages of realization and exploitation processes. The most important part of compliance analysis belongs to behavior analysis. The analysis is hold by mathematical tools before exploitation stage. In this case mathematical tools should completely correspond to OpenFlow protocol requirements.

E-nets is one of the mathematical tools that allows adequately simulate behavior of OpenFlow protocol in both case full realization and just protocol modernization. The compliance analysis of algebraic properties of E-net model vivificates functional and non-functional requirements for OpenFlow protocol realization.

Modified methods of reachability tree building for behavioral analysis of OpenFlow protocol give ability to analyze such properties as boundless, limitedness and safety and traversal the main weakness of reachability tree – appearance of symbol ω. In the modified method proposed to detect a first appearance of ω symbol, find the transition where symbol have appeared firstly and set a counter in the transition. Due to this ω symbol transforms to p_1^n, where n is the final value of the counter obtained as a result of reachability tree building. As shown in example the methods give ability to find even sequence of places that lead to the loops and deadlocks.

REFERENCES

1. Toudic JM; Linear Algebra Algorithms for the Structural Analysis of Petri Nets // Rev. Tech. Thomson CSF, 1982; 1(14): 136-156.
2. Zaitsev DA; Formal Grounding of Toudic Method. Proceedings of 10th Workshop Algorithms and Tools for Petri Nets. Eichstaett, Germany, 2003; 184-190.
3. Ichikawa A, Yokoyama K, Kurogi S; Reachability and control of discrete event systems represented by conflict-free Petri nets. Proc. of ISCAS, 2005; 85: 487-490.
4. Tkachova O, Saad I, Salim MJ; Mathematica models for analysis Software-defined network Information Technologies & Knowledge, 2015; 9(2): 111-123.
5. Murata T; Petri nets: Properties, analysis and application. Proc. IEEE, 1989; 77(4): 541–579.
6. Keller R; Vector replacement systems: formalism for modeling asynchronous systems. Computer Science Laboratory, Princeton University, Princeton, NJ, Tech. Rep., 1972; 117.
7. Tkachova O, Saad I; Method for OpenFlow protocol verification. Second International IEEE Conference «Problems of Infocommunications. Science and Technology» PICS & T-2015, 13-15 October, 2015: proc. of the conf. – Kharkiv, Ukraine, 2015; 139-140.