مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (5) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (5) 2016

تطوير نموذج تجريدي لتوصيف خصائص الشبكات المتغيرة

د. رضوان دنده*
د. قاسم قبلان**
سوسن يوسف***

(تاريخ الإيداع 1 / 8 / 2016. قُبِل للنشر في 28/ 9 / 2016)

□ ملخّص □

نظراً للعدد الكبير من قواعد النفاذ المعرفة للشبكات والتغير الديناميكي لطوبولوجيا الشبكات، فإن التحقق اليدوي من الخواص المهمة في الشبكة مثل الوصولية، عدم تضارب القواعد وعدم وجود حلقات أمراً صعباً على المبرمج.

يعدَ التوصيف الصوري(Formal Specification) للأنظمة والبروتوكولات من أهم الطرق التي تستخدم لإزالة الغموض في تعريفات الأنظمة واكتشاف الثغرات في عملها.

هناك العديد من الأبحاث التي قدمت في مجال توصيف وصولية الرزم في الشبكات لكن القليل منها تم اختبارها عبر أدوات فحص النماذج التي تساعد في كشف أخطاء هذه النماذج.

في هذا البحث تم تطوير نموذج تجريدي م ن أجل توصيف الشبكات الديناميكية ليصبح مناسباً للتحقق من مجموعة من الخصائص المهمة ومنها وصولية الرزم، عدم وجود التضاربات..الخ اعتماداً على ترميز حالة الشبكة. تم تحقيق النموذج المقترح الذي يوصف الشبكة بواسطة لغة المنطق المؤقت للأفعال (Temporal Logic of Action) +TLA, والتي هي عبارة عن لغة توصيف عالية المستوى، تعتمد على نظرية المجموعات والجبر المنطقي الأولي. تم تحليل النموذج وفحص خصائصه باستخدام أداة فحص النماذج TLC المستخدمة مع الأداة TLA، تظهر النتائج صحة النموذج وتحسيناً من ناحية تخفيض زمن استجابة وعدد الحالات المطلوبة للحصول على نتيجة التحقق.

الكلمات المفتاحية: التوصيف الصوري– الوصولية–الجبر المنطقي الأولي–TLA+

*أستاذ – قسم النظم والشبكات الحاسوبية –كلية الهندسة المعلوماتية –جامعة تشرين – اللاذقيه–سورية
**مدرس – قسم النظم والشبكات الحاسوبية– كلية الهندسة المعلوماتيه – جامعة تشرين – اللاذقية–سورية
***طالبة دكتوراه – قسم النظم والشبكات الحاسوبية –كلية الهندسة المعلوماتية – جامعة تشرين – اللاذقية– سورية

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (5) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (5) 2016

A development of an abstraction model for specifying dynamic networks properties

Dr. Radwan Dandeh[*]
Dr. Kasem Kabalan[**]
Sawsan Youssef [***]

(Received 1 / 8 / 2016. Accepted 28 / 9 / 2016)

□ ABSTRACT □

According to the large number of the access rules that define the networks, and the dynamic changing of the network topology, that is the verification by hand of the important properties in the network such as reachability, access rules conflict free and loop free is so hard to accomplish by the programmer.

Formal specification of systems and protocols is considered one of the most important methods that is used to eliminate the ambiguous of the system configurations and find bugs of its work.

A lot of the researches have been introduced in packet reachability and network specification domain, but a little of them are checked and analyzed by model checkers which help to detect the errors of these models.

In this paper an abstraction model for dynamic networks specification has been introduced and developed to be appropriate for several important properties of the network such as reachability, no conflict..etc, depending on the network state. The proposed model specification is implemented by TLA+(Temporal Logic of Action) language which is a high level specification language built on Set-theory and First Order Logic, the model has been analyzed and the properties are checked by TLC model checking tool which used by TLA tool.

Results show the correctness of the model, and improvement in reducing the response time and the required states to get the result of the verification.

Keywords: Formal Specification- Reachability-First Order Logic – TLA+

[*]**Professor, Department of Networks and Operating Systems. Faculty of Information Technology, Tishreen University, Lattakia, Syria.**
[**]**Assistant Professor, Department of Networks and Operating Systems. Faculty of Information Technology, Tishreen University, Lattakia. Syria.**
[***]**PhD student, Department of Networks and Operating Systems. Faculty of Information Technology, Tishreen University, Lattakia, Syria.**

مقدمة:

تحدد السياسة العامة للشبكة من خلال مجموعة من قوائم التحكم بالنفاذ Access Control Lists(ACLs) والتي تعرف على أنها مجموعة من قواعد النفاذ Access Rules التي تتضمن نوعين من الأفعال: السماح Permit والمنع Deny، وذلك لتحديد عملية الوصولية للرزم.

تعد وصولية الرزم وبالتالي إمكانية إجراء اتصال بين أجزاء الشبكه ومنه توافر الخدمة أمراً بغاية الأهمية في الشبكات، حيث تتأثر هذه العملية بعدة عوامل منها:

1 -توافق قواعد النفاذ المعرفة من قبل مشغل الشبكات بحيث أن أي خطأ أو تضارب في القواعد أو عدم توافقها مع المسار إلى الهدف قد يؤدي إلى سلوك الشبكة سلوكاً غير صحيح.

2 -طوبولوجيا الشبكة ومسارات التوجيه إلى الهدف.

3 -تعطل المسارات والعقد على المسار إلى الهدف.

يعد التحقق من هذه الوصولية للرزم أمراً ليس بالسهل في حال تم التحقق يدوياً من خلو الشبكة من تضاربات القواعد وحساب التوافق لوصول الرزم أو عدمها وخاصة في الشبكات الديناميكية. تعد النمذجة والتحقق الصوري من أهم الطرق المستخدمة للتحقق من توافق البرمجيات والانظمة مع المتطلبات وذلك على الرغم من أن مثل هذا النوع من عمليات التحقق لايساعد في التأكد من صحة وسلامة النظام بكل مواصفاته، لكنه يعد فعالاً في التأكد من بعض المتطلبات الحرجة، كما يمكّن من إزالة الغموض وفهم النظام بعد أن تتم كتابته بالأيدي البشرية والتي قد تفتقد التأكد من بعض الخصائص [1].

تناولت الأبحاث موضوع التأكد من الخصائص للشبكة من خلال عدة نواحي كالتأكد من خلو هذه القواعد من التضاربات، حساب وصولية الرزم وفق هذه القواعد وعملية التأكد من توافق هذه القواعد مع السياسة العامة للشبكة وخاصة في الشبكات الكبيرة، لكن هناك أبحاث قليلة تناولت تغير الوصولية مع تغير مسارات التوجيه وخروج العقد عن العمل, وتغير حالة الشبكة.

قام Guttman وآخرون بتعريف طريقة صورية لحساب مجموعه من القواعد الأمنية المعرفة على عدة أجهزة وذلك لتوصيف الحالة الأمنية العامة للشبكة، من أجل تسهيل عملية التحقق قام الباحثون بنمذجة الشبكة كمنطقة شبكة واحدة ومحدودة بأجهزة التوجيه (Routers)، هذا القرار يعكس بشكل طبيعي حالة الموجهات الداخلية حيث أنها لا تدخل في صيغه المرشح بحد ذاته.بشكل مشابه تم تعريف نموذج تدفق البيانات بشكل تجريدي، وذلك من خلال عنوان كل من المصدر والهدف ونموذج الخدمه فقط. تقوم الخوارزمية المقترحه بحساب مجموعه الرزم التي تتمكن من عبور المسار كاملاً وفقاً للقواعد الأمنية[2].

قامJeffreyوآخرون [3] بتقديم تحليل لتعريفات قواعد النفاذ باستخدام حل محدود باستخدام أداة لفحص النماذج (Model checker)، تم التركيز على خصائص الوصولية (Reachability) التي تشير إلى وصول الرزم أم لا، و (Cyclicity) التي تعني إيجاد الرزم التي لا تحقق تطابقاً مع أي قاعدة نفاذ. تم تقديم نموذج يسمح بترجمة المشكلة إلى أداة فحص النموذج(SAT-Solver) وتم إثبات أن استخدام هذه الأداة أكثر فعالية من مخطط القرار الثنائي BDD (Binary Decision Diagram)، لكن لم يتم الأخذ بالحسبان حالة الشبكة ومساراتها في هذا البحث.

تم القيام بتحليل ستاتيكي لشبكات IP بواسطة Xie وآخرين [4]،حيث تم تعريف الحد الأدنى والأعلى لتوقعات الوصوليه وفق اعتبارات قواعد النفاذ والتوجيه الديناميكي، تقوم هذه الطريقة بحساب الرزم التي من الممكن أن

يسمح بتمريرها عبر كل مسار،إذ يعرف حد التقريب الأعلى (Upper Approximation) أنها العدد الاعظمي للرزم التي من المحتمل تسليمها في الشبكة، بينما يعرف حد التقريب الأدنى (Lower Approximation) بأنه مجموعه الرزم التي سيتم تسليمها عبر كامل الشبكة تحت أي ظرف كان مع الأخذ بالحسبان تغير التوجيه في الشبكة.

قام Bera وآخرون [5,6] بتعريف إطار عمل لفلترة الرزم لكشف صحة قواعد النفاذ الموزعة وكشف الموثوقية أيضاً حيث اعتمد الباحثون على الجبر المنطقي في الحل واستخدام (SAT Slover), حيث يعطي عداداً يزداد بمقدار واحد عند اكتشاف خطأ في قواعد النفاذ وهذا يسهل كشف التضاربات في الشبكة. يتلقى النموذج وصفاً عن السياسة العامة المتبعة في الشبكة ومن ثم يقوم بفحص قواعد النفاذ في الشبكة ليرى فيما إذا كانت تحقق هذه السياسة، تتم ترجمة قواعد النفاذ إلى صيغ منطقية مع وصف السياسة العامة للشبكة بطريقة منطقية أيضاً لتتم المقارنه واكتشاف فيما إذا كانت الشبكة تحقق هذه السياسة،هذه الطريقة تقوم ببناء نموذج الشبكة وهو مخطط صوري مع قواعد نفاذ مسندة إلى الأضلاع.

في الأبحاث التي تناولت تحليل الشبكات مع الأخذ بالحسبان التوجيه في الشبكة كما في البحث المقدم بواسطة Sveda وآخرين[7]، تم استخدم جدول معلومات التوجيه (Forwarding Information Base) FIBالذي يحتوي على كافة التوجيهات التي تصل إلى كل هدف في الشبكة، كما قام الباحثون باستخد ام مفهوم الشرائح (Slices) لنمذجة الشبكة، إذ تشير الشريحة إلى عدد الشبكات التي تطبق عليها نفس سياسة النفاذ أو المرشحات (Filters). تم تخفيض التعقيد في حساب عدد الطرق الممكنة في الشبكة لكن وفقاً لمعلومات ستاتيكية من جداول التوجيه المتوفرة، فمن أجل كل حالة للشبكة يتم تحديد الطرق الفعالة التي يمكن أن تستخدم لتوجيه الرزم لكن هذه الدراسة اقتصرت على المعلومات التي يحددها (FIB) وبالتالي تعتبر معلومات ستاتيكية غير مرنة، هنا قام الباحثون باستخدام الجبر المنطقي لتقديم نموذج الشبكة ولم يتم الاختبار عملياً لمعرفه صحة النموذج كما لم يتم تناول تغير طوبولوجيا الشبكة في النموذج.

بينما تناول Ryšavý وآخرون في[8] دراسة تأثير الوصلات على حالة الشبكة حيث كان الهدف تعداد الحالات التي تنتقل إليها الشبكة عند تغير الطوبولوجيا، حيث تحدد حالة الوصلات فيما إذا كانت تعمل أو لا، الهدف من هذه الدراسة هو دراسة الانتقالات بين الحالات التي تؤول إليها الشبكة، وبدلاً من استخدام المعلومات في جداول التوجيه الحالية لتحديث الطريق المثالية في الشبكة أو الطرق المتوافرة في تلك الحالة، قدم الباحثون طريقة جديدة هي جدول الطوبولوجيا المعدلة(Modified Topology Table MTT)، والذي يتم إنشاؤه بعد حساب كافة الطرق المتوافرة في الشبكة بواسطة خوارزمية (Rubins's algorithm)، حيث تم استخدام جدول (MTT) لتحديد الطرق المتوافرة في الشبكة وذلك في كل حالة، تم استخدام المرشحات وكلفة المسار كوسائط لتحديد الحالة لكن لم يتم اختبار الحل عملياً بالإضافة للتعقيد الزمني لحساب كافة الطرق المتوافرة وعدد الحالات الكبيرة التي سيتم الانتقال إليها لتحديد نتيجة الوصولية.

لاختبار الخصائص في الشبكات الحاسوبية قام Kazemian وآخرون بدراسة الانتقالات بين أجزاء الشبكة من خلال تحليل الفضاء الممثل لترويسة الرزمة (Header Space Analysis HAS)، حيث يعتمد هذا المبدأ على اعتبار ترويسة الرزمة تتابع من البتات من خلاله يمكن استخلاص الرزم القادرة على الوصول إلى الهدف ويطبق هذا الحل على مجموعة كبيرة من الرزم، لكن الانتقالات تكون كثيرة، وبالتالي سيتولد عدد كبير من الحالات من أجل الحصول على نتيجة التحقق [9].

أهمية البحث وأهدافه:

الهدف الأساسي من هذا البحث هو تطوير نموذج صوري ديناميكي للشبكات الحاسوبية بشكل عام لتحديد الحالات التي تصلح لنقل الرزم وفق التعريفات الحالية للشبكة أي للتحقق من وصولية الرزم إلى الهدف كذلك لبيان خواص أخرى في الشبكة مثل تحديد العقد الهامة في الشبكة، عدم تضارب القواعد، حيث يتعامل هذا النموذج مع متغيرات الشبكة مثل حالة الوصلات، حالة العقد وقواعد النفاذ التي تتحكم بعملية تمرير الرزم في المسارات، بحيث يتم التجاوب مع تغير طوبولوجيا الشبكة نتيجة خروج العقد أو الوصلات عن الخدمة. تم استخدام فكرة الدمج بين مسارات التوجيه وقواعد النفاذ في النموذج المقدم في [8] لكي نقوم بتوصيف الشبكة حيث لم يتم التطبيق عملياً على أي أداة لفحص النموذج في هذا البحث والمقارنة مع النموذج المقدم من قبل Bera من خلال اختبار الحل المقدم على أداة فحص النماذج .

تم التركيز في هذا البحث على الفحص الآلي والديناميكي لوصولية الرزم وفق قواعد النفاذ، التوجيه وحالة الشبكة، والتحقق من أنها تحقق سياسة معينة وخواص مهمة للشبكة حيث يحاول الإجابة على أسئلة مثل: هل العقدة A قادرة على الاتصال بالعقدة B وفق بروتوكول معين؟ هل مخدم الانترنت متوفر للمستخدمين وفق أي حالة من حالات الشبكة؟ وغير ذلك.

يأخذ النموذج المستخدم لتوصيف الشبكة بالحسبان مايلي:

- طوبولوجيا الشبكة من عقد ووصلات ومسارات.
- حالة (الوصلات – العقد – المسارات) وتبدل حالتها.
- قواعد النفاذ المسندة للوصلات.
- إمكانية اختبار وصولية الرزم بين نقطتين وفق التوصيف السابق.
- تصميم وكتابة النموذج بطريقة ديناميكية بحيث يمكن استخدامه لاختبار العديد من السيناريوهات وإمكانية تطويره، إضافة إلى كتابته بلغة +TLA التي تصف السلوك الانتقالي للنظام بصورة سهلة. تم اختبار النموذج بأداة فحص النماذج TLC، حيث تعد عملية تقديم النموذج بهذه الأداة TLA مهمة من ناحية ديناميكية النموذج والذي يمكن المطورين من الاعتماد عليه ليصبح صالحاً لاختبار الوصولية في الشبكات المختلفة ومن أنواع خاصة مثل: الشبكات النقالة– الشبكات المعرفة بالبرمجيات..الخ.

طرائق البحث ومواده:

يعتمد هذا البحث على التوصيف الصوري الذي يبنى على ثلاثة أمور هي : التوصيف– التأكد– التحقق(Specifcation, Validationand Verification)، يشار إلى هذا المبدأ بـSV&V حيث يعد مرحلة هامة جداً في تحليل الأنظمة والتأكد من خصائصها. يتم التمييز بين المبدأين (Verification)، و (Validation)حيث يتم في عملية التحقق الإجابة على سؤال: هل هذا الشيء صحيح؟ أي هل هذا المنتج أو النظام يلبي رغبات المستخدم أما في (Validation)يتم الإجابة على سؤال: هل قمنا بالعمل الصحيح، أي هل يحاكي هذا العمل المتطلبات الحقيقية للنموذج. أما مرحلة التأكد فإنها تتألف من تعابير ستاتيكية لتوصيف البنية ومن ثم تعابير توصف السلوك (Behavior)[10].

يتم التحقق من عمل التوصيف باستخدام أدوات كشف النماذج (Model Checking) والتي هي عبارة عنتقنية تقوم ببناء نموذج منتهي الحالات للنظام واختبار مدى تحقيق النموذج لصفة معينة وذلك في كامل فضاء الحالات التي يتم توليدها، وينتهي الاختبار عند الوصول إلى الحالة النهائية حيث يعطى التعريف لأداة فحص النموذج كما يلي:

ليكن $\mathbf{M}$ نظام انتقالي معين، و F هي صيغة منطقية مؤقتة (Temporal Logic)، فإن أداة التحقق من النماذج يجب أن تقوم بإيجاد كل الحالات في فضاء الحالات للنموذج M والتي تحقق الصيغة F ويعبر عن ذلك كما يلي:

$$\mathbf{M} \vDash F \qquad [11]\ (1)$$

لتمثيل حالة الشبكة تم الاعتماد على فكرة النموذج المستخدم في البحث [8]حيث اعتمد الباحثون الجبر المنطقي الأولي (First Order logic) FOL لتمثيل حالة الشبكة، كما تم بتطوير النموذج ليصبح مخصص للوصولية وغيرها من خصائص الشبكة كعدم التضارب بين القواعد، وتحديد مكان الخطأ بدقة.لتمثيل الحالات الابتدائية والأفعال التي تؤدي إلى الانتقالات بين الحالات نستخدم النظم الانتقالية [11] (Transition Systems) و تم اعتماد النموذج الانتقالي الآتي للتوصيف حيث يتألف من:

- M = (S, I, R, L)
- مجموعة منتهية من الحالات S.
- مجموعة الحالات الابتدائية $I \subseteq S$.
- R: العلاقة الانتقالية التي تحقق الانتقالات بين الحالات $R \subseteq S \times S$.
- التابع $S \rightarrow 2^{AP}$: نظام تسمية للحالات.

حالة الشبكة يتم توصيفها بأصفار وواحدات تدل على < حالة الموجهات– حالة الوصلات >،والحالات التي قد تذهب إليها الشبكة عند تغيير الطوبولوجيا أو فشل العقد.

عند حدوث الأفعال(Actions) يتم الانتقال إلى الحالة التالية ويعبر عن ذلك رياضياً كما يلي [11] :

$$\sigma = s_0 \xrightarrow{A_0} s_1 \xrightarrow{A_1} \ldots s_n \quad (2)$$

تعتبر الأداة(TLA)Temporal Logic of Action إحدى أدوات التوصيف الصوري للنماذج المعتمدة على المنطق الرياضي حيث قام Lesli Lamport بتقديم نظرية المنطق المؤقت للأفعال والتي تمكن من شرح النموذج والقواعد المنطقية في برنامج واحد ([11], [12]).

استخدمت TLA في العديد من الأبحاث لتوصيف بعض البروتوكولات والشبكات وحل بعض المشكلات، مثلاً في[13] استخدمت لتوصيف خدمات الويب، كما استخدمت لتوصيف النظم الحساسة للزمن في [14].

تعد +TLAلغة ذات مستوى عال للتوصيف الصوري حيث تتألف من جزأين: هما منطق الأفعال، والمنطق الرياضي الأولي الذي يدعم بشكل رئيسي (Linear Temporal Logic) LTL، قد تكون صيغة +TLA صحيحة أو خاطئة، أما الفعل (Action) يكون صحيحاً بالنسبة لتصرف انتقالي ما فقط وفقط إذا كان صحيحاً لأول زوج من الحالات.

الصيغة الرئيسية لنظام التوصيف المكتوب بلغة +TLA هي[11]:

$$Init \wedge [Next]_v \wedge \mathcal{L} \qquad \textit{Liveness} \qquad (3)$$

حيث:

- *Init*: صيغة رياضية تصف حالة البداية.
- *Next* : صيغة رياضية تصف الانتقالات بين الحالات.
- *Liveness :* هي صيغة لضمان عدم بقاء النظام بحالة واحدة.

تعتمد هذه اللغة على معاملين هامين هما (always)□والذي يعني دائماً، و (eventually) ◇ والتي تشير إلى أن ورود هذا الحدث بعد حدث معين في المستقبل قد يكون دائماً.

الغاية الرئيسية من اختيارنا لهذه الأداة أنها لغة غنية ببنى المعطيات وليست كغيرها من الأدوات التي تعتمد على علاقات بسيطة لتحليل النظام، بينما في +TLAيمكن تمثيل حالة النظام بالكثير من المتغيرات والعلاقات بين الحالات بطريقة دقيقة.الميزة الهامة أيضاً لهذه الأداة هي استخدام الوحدات (Modules) فهي تحتوي على جزء معرف يمكن اشتقاقه وجزء يتم كتابته ويتم إيراده واشتقاقه وهذا يجعل النموذج قابلاً للتعديل والتطوير من قبل باحثين آخرين والاستفادة من النموذج المكتوب.

تعدTLC أداة فحص النموذج المترافقة مع +TLAأداة هامة تمكن من اكتشاف الأخطاء في النموذج وتعداد الحالات التي يتم الانتقال إليها، كما يمكن استخدامها لإثبات النظريات المتعلقه بالنظام واختبار خصائص مكتوبة بالجبر المنطقي ليصار إلى تحديد الحالات التي تم انتهاك الخصائص عندها.

السياق الذي تكتب به [11]+TLA:

$$\langle formula\rangle \triangleq \langle predicate\rangle \mid \langle formula\rangle \wedge \langle formula\rangle \mid \neg\langle formula\rangle \mid \Box[\langle action\rangle]\langle state\ function\rangle \mid \Box\langle formula\rangle \mid \Diamond\langle formula\rangle \tag{4}$$

1 -النموذج الصوري المقترح للشبكة:

تم في هذا البحث تمثيل الشبكة الحاسوبية بمخطط بياني موجه حيث تشير العقد فيه إلى أجهزة الشبكة بينما تشير الأسهم إلى الوصلات وطريقة اتصال هذه العقد مع بعضها البعض ويتم إسناد قواعد نفاذ للأضلاع كما في الشكل (1)، وذلك لكي نختصر عملية التمثيل الرياضي بدون استخدام المنافذ. هذا المخطط قابل للتغير وفق حالة الشبكة فقد تتعرض الوصلات للانقطاع وتغير في الطوبولوجيا نتيجة فشل الموجهات أو الوصلات.

تم توصيف نموذج الشبكة باستخدام الجبر المنطقي الأولي (FOL)، وذلك في أول مرحلة ليصار إلى ترجمتها فيما بعد إلى لغة منطقية ليتم اختبار العمل فيها.

تم تمثيل الشبكة كما يلي:

$$N = (R, L, F) \tag{5}$$

حيث:

- R إلى مجموعة منتهية من الأجهزة في الشبكة .
- $L \subseteq RxR$ مجموعة منتهية من الوصلات بين الأجهزة.

- F هي مجموعه منتهية من قواعد النفاذ يتم إسنادها لكل وصلة في مخطط الشبكة. وسنعتمد التمثيل الآتي للقاعدة: المصدر–المستقبل–نوع الاتصال، والأفعال: سماح أو منع، كما مبين في الجدول (1).

الجدول (1) مثال عن قواعد النفاذ

	src	dst	proocol	action
R_1	214.0.0.0/8	*	TCP	permit
R_2	214.65.0.0/16	*	UDP	deny

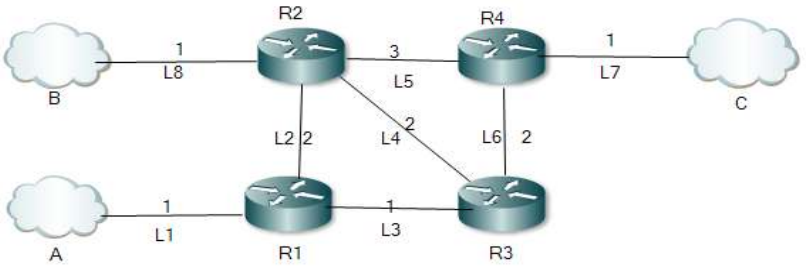
يشير الرقم المسند إلى الوصلة إلى قوائم النفاذ المطبقة، مثال:

1: permit HTTP any any

Deny UDP B A

2: permit HTTP B any

3:Permit any any any



الشكل (1) طوبولوجيا شبكة تصل بين عدة شبكات

لتمثيل عناوين IP بشكل منطقي فإنها تعرف بأربع رموز [8]:

$$IP = \{a_1, a_2, a_3, a_4 : a_i \in Seq(0..255) : i \in \{1,2,3,4\}\} \quad (6)$$

ومن أجل تمثيل قناع الشبكة الفرعية $10.10.10.0/24$ يعطى التمثيل الرياضي بالشكل الآتي [8] :

$$IPN = \{< a, m >, a \in IP \wedge m \in \{0.32\}\} \quad (7)$$

أما بالنسبة للرزم فإنها تتألف من الترويسة (Header) والحمولة (Payload) ، تعتمد عملية توجيه هذه الرزم على المسار الموجود في جدول التوجيه حيث لابد من تمثيل المصدر والهدف ونوع الرزمة لمعرفة المسار الذي يجب توجيهها فيه، هنا سيتم تحويل هذه التوابع الرياضيه إلى لغة +TLA لذا نقوم بتقسيم نوع الرزم إلى عدة أنواع ونضعها في مجموعة:

$PackType = \{a : a \in \{IP, ICMP, TCP, UDP, HTTP\}$
$SrcAdd = \{s : s \in IPN\}$
$DestAdd = \{d : d \in IPN\}$
$Packet = \{< a, s, d >: a \in PackType, s \in SrcAdd, d \in DestAdd\}$

تعريف1 : المسار هو مجموعه من الموجهات والوصلات التي تصل بين الموجهات، من الممكن أن يكون هناك عدة مسارات تصل بين الموجهات ويعتمد اختيار المسار على بروتوكول التوجيه الذي اعتمد معياراً معيناً لاختيار الأفضل من بين المسارات، يتم التعبير منطقياً عن المسار كما يلي[8] :

$$\pi_{0n}^{k} = (R_0 l_{01} R_1, \ldots\ldots, R_{n-1} l_{n-1\ n} R_n) \quad (8)$$

تعريف2 : نعدل على توصيف حالة المسار بحيث يتم حسابها من خلال الأخذ بالحسبان حالة الوصلات المكونة للمسار، وتعطى بالعلاقة الآتية:

$$N_s^{\pi} = S_{l_{01}} \wedge \ldots\ldots \wedge S_{l_{(n-1)(n)}} \quad (9)$$

حيث تعني تقاطع حالات الوصلات $S_{l_{ij}}$ المكونة للمسار.

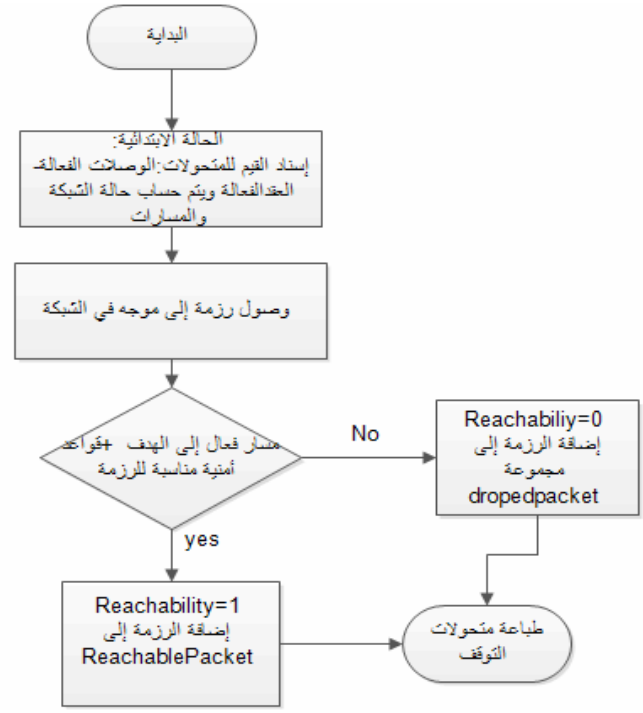
تعريف3:حالة الوصلات المكونة للمسار يتم حسابها كما يلي:

$$S_{l_{ij}} = n_s^{r_i} \wedge n_s^{r_j} \wedge LS_{l_{ij}} \quad (10)$$

حيث تشير $n_s^{r_i}$ إلى حالة الأجهزة وتأخذ القيمة صفر في حال إغلاق الجهاز وواحد في حال عمله، وبالتالي فإن حالة الوصلة تتم من خلال حساب تقاطع حالة الأجهزة الطرفية للوصلة مع حالة الوصلة فيزيائياً، هل هي مقطوعة أو متصله.

من أجل توجيه الرزم يجب معرفة سياسة النفاذ العامة في الشبكة، وتمثيل القواعد التي تتألف منها قائمة النفاذ.

نهدف هنا إلى تمثيل الشبكة وفق حالات مكونة من حالة الوصلات، يتم ترميز الحالة رقم 0 أو 1 لكي يتم تسهيل عملية المقارنات المنطقية.في هذا البحث يتم مراعاة التغيير الديناميكي في الشبكة نتيجة فشل العقد والمسارات لا بد من أخذه بالحسبان لمعرفة الحالة التي ستؤول إليها الشبكة كما هو موضح في الشكل(2).



الشكل(2) مخطط النموذج المقترح لتحليل الوصولية في الشبكة

المدخلات : عدد العقد– عدد العقد الفعالة– عدد الوصلات وعدد الوصلات الفعالة– قواعد النفاذ.

التوابع : إسناد قواعد النفاذ للوصلات وتكوين حالة الشبكة.

الإجراءات: خطأ في الوصلات أو العقد– هل رزمة ما قابلة للوصول إلى الهدف وفق الحالة الراهنة للشبكة.

4-1 توصيف النموذج باستخدام +TLA

تعريف الثوابت والمتحولات في الشبكة:

$$\boldsymbol{CONSTANTS}\ Node, An, Edge, Ae, Data, \quad (11)$$
$$Rules, NetIP\ \boldsymbol{VARIABLE}\ NodeStatus, EdgeStatus, PathState, packet, Reachability, Netr, EdgeAcl$$

الثوابت هي عبارة عن مجموعة العقد *Node*، ومجموعة الوصلات *Edge*، والوصلات الفعالة *Ae* التي تعد حالتها 1، وعدد العقد والعقد الفعالة *An* التي تعمل وحالتها 1، كما يجب تعريف قواعد النفاذ في الشبكة والمرتبطة بكل وصلة والتي سيتم إسنادها من خلال ملف التعريف المستخدم لفحص التوصيف المكتوب بـ+TLA.

من أجل تعريف المخطط الصوري للشبكة من خلال الثوابت والمتحولات السابقة نستخدم التابع الآتي الذي يعرف المسارات في الشبكة:

$$Path \triangleq \{p \in Seq(Node): \wedge p \neq <> \wedge \forall i \in 1..(Len(p)-1): < p[i], p[i+1] > \in Edge \quad \}(12)$$

في ملف التوصيف في +TLAنقوم بتسمية الموجهات والأضلاع بمحارف بدلاً من الأرقام والعناوين وذلك لتسهيل المقارنات إذ نمثل قاعدة النفاذ الافتراضية بفراغ " " للإشارة إلى أن القيمة هنا لاتهم مثلاً للقيام بتمرير جميع الرزم بدون الاهتمام للمرسل أو إذا كانت في مسقط المستقبل فلا يهمها المستقبل وهكذا..

4-1-1 تعريف التوابع الأساسية:

يقوم التابعEinP بعملية اختبار فيما إذا كانت وصلة معينة تنتمي إلى مسار P.

$$EinP(e \in Edge, p \in Path) \triangleq \exists\, i,j \in 1..Len(p): \quad (\ll p[i],p[j] \gg = e) \vee (\ll p[j],p[i] \gg = e) \tag{13}$$

لمعرفة ما إذا كان هناك عقدتين متصلتين نختبر ذلك بالتابع التالي:

$$AreConnectedIn(m,n) \triangleq \exists\, p \in Path: (p[1] = m) \wedge (p[Len(p)] = n) \tag{14}$$

نختبر أيضاً فيما إذا كانت عقدة ما تنتمي لضلع معين:

$$Nin(n,e) \triangleq (n = e[1]) \vee (n = e[2]) \tag{15}$$

قمنا بكتابه هذه الإجراءات من أجل تحديد الحالة الجديدة التي ستأخذها الشبكة في حال فشل الوصلة أو فشل عقدة.

تم تمثيل قاعدة النفاذ بشكل :

<source, Destination, ProtoType, Action>، حيث Action يكون إما السماح أو المنع وذلك وفقاً للفرض التالي:

$$ASSUME\ Rules \subseteq (Node \times Node \times \{\text{"HTTP"},\text{"IP"},\text{"UDP"},\text{" "}\} \times \{\text{"permit"},\text{"deny"}\}) \tag{16}$$

لتعريف المتحولات تمت الإشارة إلى حالة العقد والوصلات والطرق بأصفار وواحدات، كما تم إدخال حالة الرزمة لكي تكون إما قابلة للوصول للهدف أو غير قابلة.

$$TypeEnvirent \triangleq \wedge NodeStatus \in [Node \longrightarrow \{0,1\}]$$
$$\wedge EdgeStatus \in [Edge \longrightarrow \{0,1\}]$$
$$\wedge PathState \in [Path \longrightarrow \{0,1\}]$$
$$\wedge packet \in \begin{bmatrix} Source: Node, Dest: Node, D: Data, State: \\ \{\text{Reachable},\text{NotSEnd}\} \end{bmatrix} \tag{1}$$
$$\wedge Reachability \in \{0,1\}$$
$$\wedge Netr \in [Node \setminus X\ Node \longrightarrow \{0,1\}]$$
$$\wedge EdgeAcl \in [Edge \longrightarrow SUBSET\ Rules]$$

التابع الذي يعطي حالة الوصلة:هو عبارة عن التقاطع المنطقي لحالة العقد على طرفي الوصلة مع حالة الوصلة بحد ذاتها هل هي مقطوعة أم لا؟تتخذ الحالة البدائية الوضع التالي:يتم إسناد قيمة 1 لكل العقد الفعالة ولكل الوصلات الفعالة، ومن ثم يتم حساب حالة الوصلات جميعها وفقاً لهذا الإسناد من خلال X(c):

$$X(c) \stackrel{\Delta}{=} \forall i \in Len(c-1): \wedge NodeStatus[c[i]] = 1$$
$$\wedge NodeStates[c[i+1]] = 1 \tag{18}$$
$$\wedge EdgeStatus[< c[i], c[i+1] >] = 1$$

في الحالة البدائية تتم عملية إسناد القيم للمتحولات ومن ثم تحصل الأفعال التي ذكرناها سابقاً لكي يتم توليد الحالات. لحساب وصولية الرزم من نوع HTTP، والذاهبة من العقدة المصدر 1 إلى الهدف يجب وجود مسار فعال بين هاتين العقدتين بحيث تصبح القيمة الجديدة للمتحولReachable هي 1.

فيما يلي عملية الإسناد لبعض التوابع لكي تأخذ المتحولات قيمها، حيث تشير التعابير الآتية إلى اختبار الحالة التالية الناتجة عن الحالة الراهنة والتي نريد معرفة من خلالها توقع وصول الرزمة.

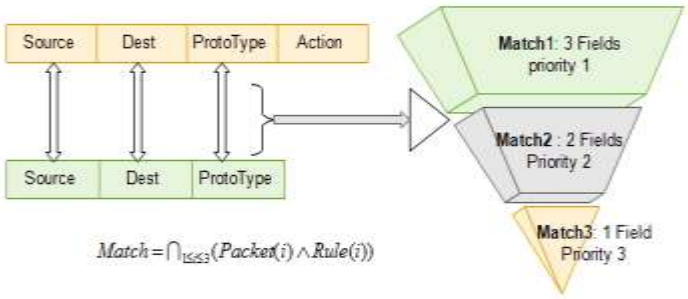
$$Init \triangleq \wedge NodeStatus = [i \in Node \mapsto IF\ i \in An\ THEN\ 1\ ElSE\ 0]$$
$$\wedge EdgeStatus = [b \in Edge \mapsto IF\ b \in Ae\ THEN\ 1\ ELSE\ 0]$$
$$\wedge PathState = [c \in Path \mapsto IF\ X(c)\ THEN\ 1\ ESLE\ 0] \quad (19)$$
$$\wedge Reachability = 0$$
$$Send(\alpha, \beta) \triangleq \wedge\, \alpha[1] = \beta[1]$$
$$\wedge\, \alpha[2] = \beta[Len[\beta]$$
$$\wedge\, PathState[\beta] = 1$$
$$\wedge\ Rea(\alpha, \beta)$$

$$next \triangleq \exists\ t \in Path : Send(Packet, t)$$
$$Spec \triangleq Init \wedge \square[next]_{<EdgeStatus, NodeStatus, PathStat, packet, Reachability>}$$

من أجل أن تتغير قيمة المتحولReachableإلى 1 يتم اختبار فيما إذا كان يوجد مسار له المصدر والمستقبل الموجود في الرزمة وحالة هذا المسار1 وتسمح لها السياسة الأمنية للمسار بالعبور.

- **طريقة تحديد التطابق بين قواعد النفاذ والرزم:**

هنا قمنا باستخدام تطابق هرمي بين قواعد النفاذ , وترويسة الرزمة، وذلك لمنع حدوث لبس وغموض وتخفيف التعقيد قدر الإمكان كما هو موضح في الشكل(3):



الشكل (3) عملية التطابق بين الرزمة وقواعد النفاذ

تدل هذه التوابع على توافق قاعدة النفاذ مع مساقط الرزمة بالترتيب أي المصدر مع المصدر والمستقبل مع المستقبل ونوع الرزم حسب ترتيب وروردها في تعريف الصيغة.

هنا تم تقسيم التطابق إلى ثلاثة توابع هي *Match1*, *Match2* , *Match3*، حيث يعني إذا تطابقت الرزمة بالكامل مع قاعدة نفاذ فهذه تعني أولوية عليا وإذا لم يتحقق التطابق نقوم بإنقاص عدد حدود التطابق في كل من*Match1*و *Match2* ومن ثم يتم تطبيق تابع *Match* كما يلي:

$$Match(Pac \in Packet) \triangleq \exists i \in Rules : \vee \wedge Mtach1(Pac, i)$$

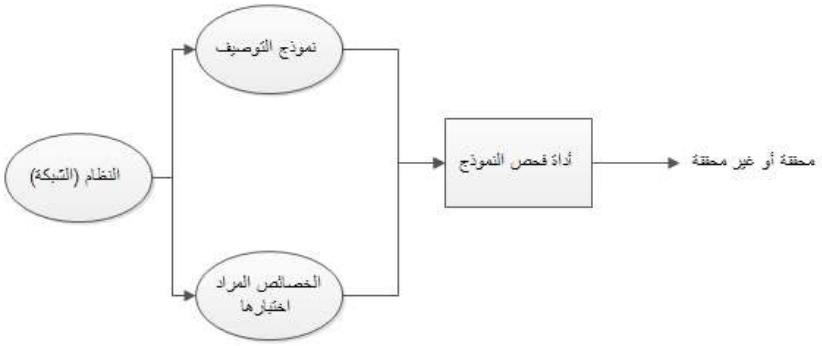
$$\vee \wedge \sim Match1(Pac,i) \wedge Match2(Pac,i) \quad (20)$$
$$\vee \wedge \sim Match1(Pac,i) \wedge \sim Match2(Pac,i) \wedge Match3(Pac,i)$$

كما نرى من الصيغة السابقة فإن اختبار عملية الوصول للرزمة وفق قاعدة النفاذ يجب أن تكون مطابقة بالكامل للقاعدة والفعل الموافق هو السماح بالمرور (permit), وفي حال عدم التطابق وفق الحقول يتم الانتقال إلى تابع التطابق الثاني الذي يختبر الحقلين التاليين وذلك في حال عدم تحديد الاختبار وفقاً للصيغة : أيا يكن" "، وعند عدم التحقيق يتم الانتقال إلى التابع الثالث للتطابق وذلك بتغيير الحقول،تضمن هذه الطريقة أن لاتمر رزم غير مسموح بها وفق قاعدة نفاذ، فإذا حصل التطابق فإنه سيكون وفق أفضل أولوية ولكي تمر الرزمة يجب أن يكون $i[4] = \text{"permit"}$ للقاعدة المطابقة.

النتائج والمناقشة:

تعدTLC أداة للتحقق باستخدام نظرية الأوتومات للتوصيف المكتوب بواسطة +TLA، حيث تم تطويرها من قبل الباحثين الذين عملوا بلغة +TLA. تعتمد هذه الأداة على فحص صحة النموذج وفق ملف خصائص التعريفات التي سيتم إسنادها للصيغ الرياضية. عند تنفيذ الفحص يتم إعطاء الحالات التي يتم الوصول إليها وفقاً لهذه التعريفات أو مايسمى بـ (Reachable states)، وذلك اعتباراً من الحالة البدائية.

تستمرTLC بإنتاج الحالات التي يمكن الوصول إليها حتى الوصول إلى حالات تنتهك الشروط عندئذ تطبع متحولات السياق (Trace) الذي تم التوقف عنده. يبين الشكل(4) طريقة الاختبار بأداة فحص النماذج [11].



الشكل(4) كيفية اختبار الخصائص في أداة فحص النموذج

لا تنفذ الأداة TLC الشروط وفق ملف التعريفات إذا كانت القيم المسندة لاتوافق الفرضيات المنطقية والتمثيل الرياضي المكتوب في ملف التوصيف(Specification File).

سنختبر النموذج الموصف في الفقرة 4 على الأداة TLC لكي يتم التأكد من صحته، بداية يمكن بوساطة الأداة TLA أن يتم التأكد من أن الصيغ المنطقية لاتحتوي على تعارض منطقي ولا يحتوي النموذج على تعريفات متناقضة منطقياً، ويتم ذلك من خلال الأمر Parse module في الأداة وذلك قبل أن يتم اختباره على أداة TLC.

تختبر النموذج على حالتين: الحالة الأولى وجود مسار واحد بين المصدر والهدف يقبل توجيه الرزم. الحالية الثانية وجود عدة مسارات توافق قواعد النفاذ عليها عملية تمرير الرزم وذلك لمعرفة مدى مقدار تزايد عدد الحالات التي يتطلبها النموذج لإعطاء النتيجة وزمن استجابة النموذج.

كيفية الحصول على النتائج:

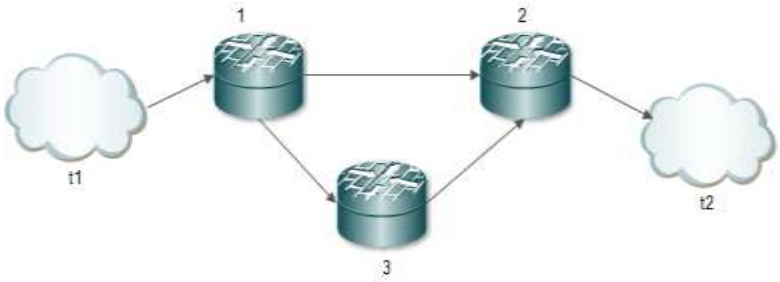
يتم تحويل تعريفات الشبكة إلى القيم التي يفهمها النموذج بعد ذلك يتم إسناد القيم للمتحولات المعرفة بالعلاقات في الفقرة 4، كما يتم تحديد الخصائص المطلوب التأكد من تحقيقها في الشبكة: تحديد الوصولية يتطلب اختبار عدة تعابير وقضايا منطقية (العلاقة 19 – الإجراء send) وذلك لاختبار وجود مسار عقده ووصلاته فعالة، كما تتوافق قواعد النفاذ المعرفة على هذا المسار مع الرزمة الواردة والسماح لها بالمرور)، لذلك يتم الانتقال بين عدة حالات (عند تغيير المتحولات الابتدائية يتم الانتقال إلى حالة جديدة واختبار قيم جديدة في الموقع الجديد). تقوم TLC بإظهار تغير المتحولات خلال تنفيذ النموذج وعند اكتشاف أن الرزمة مثلاً غير قابلة للوصول للهدف وفق الفرض المدخل من قبل المبرمج عندئذ تقوم بطباعة المتحولات وسلسلة الاحداث التي أدت إلى هذا الخطأ، وبالتالي يكون المبرمج قادراً على تحديد كان الخطأ في تعريفات القواعد أو تحديد الوصلة أو العقدة الخارجة عن العمل.

تطبع TLC عدد الحالات والزمن اللازم للحصول على النتيجة.

- **السيناريو الأول:**

يبين الشكل(5) طوبولوجيا الشبكة التي تم اختبار النموذج عليها وذلك للتأكد من أن النموذج أثناء التنفيذ لا يتعرض للإقفال والتأكد ثانياً من أن الشبكة بتعريفاتها وطوبولوجيتها الحالية تحقق السياسة العامة للشبكة:

في الحالة الأولى اخترنا شبكة صغيرة مكونة من ثلاث عقد ومسارين فقط للوصول إلى الهدف وقواعد نفاذ لتحقيق السياسة العامة للشبكة: لايجب أن تذهب الرزم بين t1 و t2 والتي هي رزم HTTPعبر الموجه 3، ولا يوجد خطأ في تعريف القواعد في هذه الشبكة.



الشكل (5) طوبولوجيا الشبكة في السيناريو الأول

ولتحقيق ذلك وضعت قواعد نفاذ كما يظهر ملف التعريف (**Configuration file**) الذي يتم إدخاله إلىTLC:

CONSTANTS

Switch = {1,2,3}

Edge={<<t1,1>>, <<1,2>>,<<2,t2>>, <<1,3>>}

As={1,2,3}

Ah={t1,t2}

Rules= {<<t1,t2,HTTP,permit>>,<<t1,t2,Ip,permit>>,<<t1,t2,UDP,permit>>, <<t1,t2," ",permit>>,,<<t1,t2,Ip,permit>>,<<t1,t2," ",deny}

INVARIANT TypeInvariant

SPECIFICATION Spec

حيث يتم إسناد متحولات للصيغ والتوابع الرياضيه المستخدمة في ملف التوصيف، حيث يحتوي السطر الأول Constants على ثوابت نموذج الشبكة المبين في الشكل(4) بينما يحتوي السطر الثاني على القيم الابتدائية التي سيقوم النموذج بإسنادها للمتحولات وفق الصيغة TypeInvariant المعرفة مسبقاً في العلاقة (17)، وبعدئذ تقوم الأداة TLC بمعالجة الصيغة الرياضيه Spec[11] والتي تبدأ من حالة بدائية وتتوقف عند حالة نهائية وفقاً للقيم الموجودة في الملف لإنتاج الحالات الصحيحة والانتقالات بين الحالات حتى الوصول إلى حالة توقف نتيجة نقض إحدى الخصائص والتي هي على سبيل المثال 0 =*Reachability*.

$$Spec \triangleq Init \wedge \square[next]_{<EdgeStatus,NodeStatus,PathStatus,PathState,packet,Reachability>} \quad (20)$$

هذا التعبير يفسر كما يلي: بدءاً من الحالة الابتدائية أوجد الحالات التالية التي تتغير فيها قيم المتحولات: حالة العقد الوصلات والمسارات والرزم والمتحول Reachability الذي قمنا بإسناد قيمة أولية له هي 0،ومن ثم يتم تنفيذ النموذج للتحقق من الصحة وذلك للزرمة التي مصدرها t1 وهدفها t2 ونوعها HTTP، حيث يتم إعطاء (parsing) في الأدة لإظهار أن الملف مكون من تعابير صحيحة ومتوافقه مع بعضها البعض رياضياً ولايحتوي على تعريفات متناقضة، ومن ثم تنفيذ النموذج. تم إنتاج حالتين وتغيير (Reachability) من 0 إلى 1 وبزمن 0.02s. وهذا يعني أن الرزمة وفق هذه الطوبولوجيا ووفق حالة الشبكة الراهنه وحالة قواعد النفاذ الموجودة والمسندة للوصلات هي قابلة للوصول للهدف.

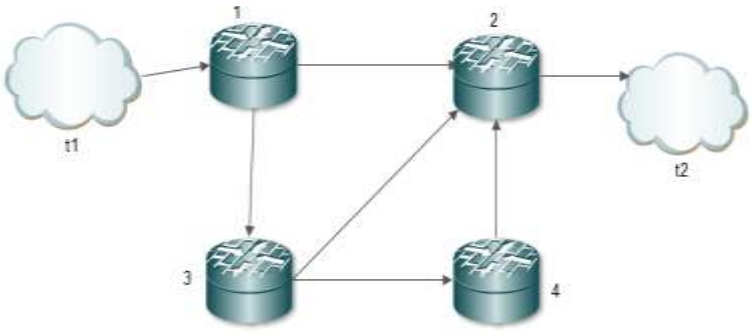
يتم التنفيذ إما من خلال نافذةTLC أو من خلال الطرفية (Console) من خلال الأمر التالي:

Java tlc2.TLC Filters.tla – deadlock (22)

يقوم الأمر السابق باختبار خاصية عدم وجود إقفال للنموذج Filters.tlaوفق ملف التعريف الذي يأخذ اسماً افتراضياً هو (Filters.CFG).

- **السيناريو الثاني:**

باستخدام عدد أكثر من العقد (4 عقد) كما في الشكل (6) حيث تم استخدام 14قاعدة نفاذ (كل وصلة قاعدتان) واختبار النموذج في حال وجود عدد من المسارات بين المرسل والمستقبل وليس مسار واحد وعدد وصلات أكثر وذلك لتحليل الزمن وعدد الحالات التي سيأخذها TLCللوصول إلى نتيجة التحقق. حيث تعاني عملية التوصيف والاختبار بأداة فحص النماذج من مشكلة انفجار الحالات (state explosion) في حال تزايد العقد أو القيم التي يتم فحصها، والتي تتميز +TLA بأنها تخفف منها بشكل كبير[11].



الشكل (6) السيناريو الثاني (قواعد أكثر+ عدة مسارات للهدف+ حقن للخطأ)

تم إسناد قواعد نفاذ مختلفة للأضلاع بحيث يتم توزيع الرزم لكي تعبر جميعها ضمن مسار واحد والذي هو المسار الأقصر <<t1,1,2,t2>>واستخدامه فقط لرزم UDPوبالتالي يجب إضافة خصائص للتحقق من صحة قواعد النفاذ حيث يتم إسناد $acl1$لوصلات المسار <<t1,1,2,t2>>:

$$acl1 = \{permitUDPfromt1\ tot2$$

$$denyanyfromt1\ tot2\}$$

سياسة النفاذ المسندة لبقية الاضلاع هي:

$$acl2 = \{permitanyfromt1\ tot2$$

$$denyUDPfromt1\ tot2\}$$

للتأكد من الرزم من نوع UDP ستعبر المسار الأقصر نضيف متحول للاختبار سنسميه h حيث يعطي المسار الذي سلكته الرزمة للوصول إلى الهدف وفق القواعد المتاحة , ويتم إضافة الصيغة الآتية ليتم الإسناد عند الحالة الابتدائية:

$$h \in [packet \longrightarrow path] \quad (23)$$

وعند استقبال الرزمة للهدف يتم إسناد المسار إلى هذا المتحول $h' = t$وبالتالي نضيف الخاصية التي نريد للنموذج أن يحققها، (لايوجد رزم UDPتمر بالموجه 3) :

$$NotPassPropperty \triangleq \wedge packet[3] = \text{UDP} \quad \wedge \sim in(3,h) \quad (24)$$

حيث يعرف التابع inليختبر فيما إذا كانت عقدة معينة تنتمي لمسار معين كما يلي:

$$in(node,path) \triangleq \exists\, i \in 1\,..\,(Len(path)-1):\ path[i] = node \quad (25)$$

عند التنفيذ على هذا النموذج كانت النتيجة صحيحة بدون انتهاك وبالتالي التعريفات صحيحة ولاتوجد تضاربات، إذ أن قيمة المتحول h كانت المسار التالي: {<<t1, 1, 2, t2>>} أما في حال استخدام رزم غير UDP أعطت قيم المتحول h مايلي: {<<t1, 1, 3, 2, t2>>}وبالتالي لم تسلك المسار الذي يمر بين " 1" و " 2" وهو المطلوب.

تم أيضا التحقق وإنتاج حالتين فقط وبزمن 0,02s وبالتالي وفق التوصيف المقدم في هذا البحث لا يوجد مشكلة انفجار الحالات، لذا قمنا من خلال التوصيف في هذا النموذج (الفقرة 4) على تقليل عدد الحالات الناتجة من الانتقالات بين الأفعال حيث تم التعديل على طريقة تحديد الوصولية من خلال فعل واحد حيث تم اختبار الوصولية من خلال شرط منطقي على كامل المسار(العلاقة 17)، وليس اختبار الشروط عند كل انتقال للرزمة وبالتالي لايسبب تغيير حالات عند كل انتقال للرزمة كما تمت معالجتها في الأبحاث السابقة مثل [4] و[9].

وفق النموذج السابق يمكن تحديد المبدلات أو الأهداف التي يمكن لمبدل الوصول إليها بالنسبة لرزمة معينة أو تطبيق معين، ويعبر عن ذلك بالتعريف الآتي:

$$NetReachable(R,P) \triangleq \{R' \in Node : \exists \tau \in Path : \tau[1] = R \wedge \tau[len[\tau]] = R' \wedge X[\tau] = 1 \wedge \Phi(R,R') = True$$

(26)

إذ تعد العقدة أو الموجه R’ من العقد التي يمكن لـ R الوصول إليها إذا ماوجد مسار مصدره R ومستقبله R’وحالة الوصلات ضمنه فعاله، مع توافق قواعد النفاذ والسماح بالمرور للرزمة من نوع P.

من أجل كشف التضارب في تعريف قواعد النفاذ نعرف التابع التالي الذي نقوم باختباره عند كل انتقال للرزمة وذلك لاكتشاف فيما إذا كان هناك عدة تطابقات للرزمة وبذات الأولوية:

$$MisConfig(p \in Packet, e \in Edge) \triangleq \exists p \in Packet:$$
$$\exists m, n \in EdgeAcl: \vee Match(m,p) \wedge Match(n,p) \vee Match1(m,p) \wedge Match1(n,p)$$
$$\vee Match2(m,p) \wedge Match2(n,p) \quad (27)$$

يكتشف هذا التابع خطأ في حال اكتشاف تعريفات خاطئة ومتناقضة للقواعد مثل:

Permit HTTP any any

Deny HTTP any any

كما نرى فإن هذا النموذج يتميز بالديناميكية حيث أن التوصيف قابل لإضافة أي شروط أو خاصة يريد المستخدم أن يتحقق منها على الشبكة وذلك من خلال كتابة الصيغة في التوصيف أو في مربع invariant (formula) التي توجد ضمن أداة TLC.

من الممكن أن تتعرض الشبكة لأخطاء تسبب فشل العقد– الوصلات وبالتالي تصبح مسارات معينة غير صالحة للاتصال، وفي هذا النموذج عند الخطأ تتحول حالة العنصر الذي أصبح خارجاً عن العمل إلى صفر ويزال من مجموعة الأجهزة الفعالة الذي ينتمي إليها مثل تعطل عقدة وخروجها عن العمل:

$$switchFailuer(s) \triangleq \wedge SwitchStatus[s] = 0$$
$$\wedge \forall n \in Ae :$$
$$\wedge ine(s,n) \quad (28)$$

$$\wedge EdgeStatus' = [EdgeStatus\ EXCEPT\ ![n] = 0]$$
$$\wedge UNCHANGED\ [rest\_vars]$$

وفق التوصيف المقدم فإن تعطل إحدى العقد سيحول حالتها إلى صفر وإزالتها من مجموعة العقد الفعالة Ae، وتحويل حالة الوصلات التي تمر بها إلى صفر أيضاً وذلك من خلال التابع $ine(s,n)$.

$$ine(s,n) \triangleq \vee (n[1] = s) \vee (n[2] = s)$$
(29)

وبالتالي عند ورود الرزمة وحساب الوصولية لها يؤخذ بالحسبان أن المسارات أصبحت حالتها صفراً لاتستطيع الرزمة المرور بها ويقوم النموذج بالبحث عن مسارات أخرى.

- **مقارنة مع Sat–Solver:**

في الجدول (2) سنقوم بمقارنة بعض القيم التي قام الباحثون بنشرها وفق البحث المقدم من Bera[5] وآخرون على اعتبار أنه البحث الذي تم فيه الاختبار على إحدى أدوات فحص النماذج Sat–Slover.

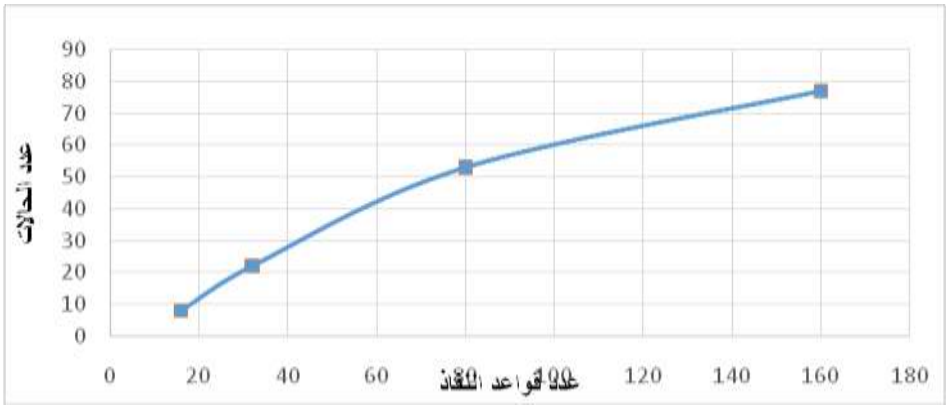
الجدول (2) مقارنة زمن الاستجابة لبعض الخصائص للنموذج المقترح مع النموذج في [5]

رقم السيناريو	عدد قواعد النفاذ	عدد قواعد النفاذ الخالية من التضارب	زمن الاستجابة(Sec) Sat–solver	زمن الاستجابة (Sec) TLA
1	10	10	7.16	0.03

2	23	25	8.17	0.03
3	38	38	8.34	0.1
4	45	94	9.33	1.02

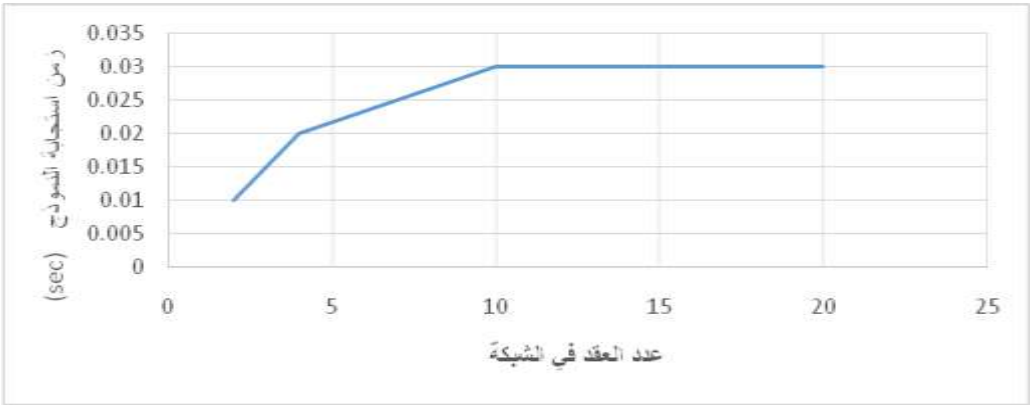
حيث نلاحظ انخفاضاً بالزمن الذي يتطلبه النموذج للوصول إلى النتيجة على اعتبار أن الاداة تقوم بكشف الحالة الخاطئة وليس تعداد الحالات الصحيحة.

باختبار عدد أكبر من العقد الواصلة بين المصدر والمستقبل وبالتالي زيادة عدد قواعد النفاذ لدراسة أثر عدد العقد في هذا النموذج على الحالات اللازمة للاختبار الشكل(7)، والزمن الشكل(8)، فإننا نلاحظ تزايد عدد الحالات التي يتم إنتاجها من قبل أداة فحص النماذج لكي يتم إنتاج المطلوب لكن تبقى ضمن الحدود المقبولة.



الشكل(7) النتائج في أداة TLC العلاقة بين عدد قواعد النفاذ في الشبكة وعدد الحالات التي يتطلبها النموذج لتحديد الجواب

بالتالي فإن النموذج لم يؤدي إلى مشكلة انفجار الحالات وذلك بسبب ميزة +TLA والتي تختلف عن النظم الانتقالية بإسناد القيم للمتحولات على كامل السلوك (عدد الحالات) بالإضافة إلى اختيار الحالة الخطأ وليس كل الحالات الصحيحة.



الشكل(8) نتائج تطبيق النموذج وفق أعداد مختلفة للعقد مقابل زمن الاستجابة للنموذج بالثانية

يبين الشكل (8) أثر تزايد عدد العقد في الشبكة على الزمن اللازم للاختبار وهي مقبولة وصغيرة مقارنة بالزمن الذي يتزايد بشكل كبير وفق الأدوات الأخرى والنماذج الصورية التي تم اختبارها على أدوات فحص النماذج الأخرى كما

في الأبحاث في [15] حيث تراوح الزمن لمعالجة أي ثغرة Bug 0.07 و0.18sوذلك على عدد عقد 2 و3 و4. وعن [16] والذي تم الاختبار فقط على عقدتين، مثلاً عملية التحقق من عدم وجود حلقات في هذه الطوبولوجيا يتطلب 195 حالة خلال 0.1 second مع العلم أنه تم التنفيذ على جهاز بمعالج 2.40 GHz Intel Core 2 Quad أما في هذا البحث فقد استخدمنا جهاز بمعالج 1,8 GHz Intel Core i3.

(يمكن أن يتم تخفيض الزمن اللازم لإنتاج نفس العدد من الحالات من خلال توزيع العمل للنموذج على عدة معالجات أو باستخدام معالج أسرع لكن هذا لن يغير بعدد الحالات لأن ذلك يعتمد على طريقة معالجة النموذج للمشكلة وعدد المتحولات التي يستخدمها).

وبالتالي نلاحظ من التجارب السابقة:

السيناريو الأول: وجود مسار واحد يحقق توافق القواعد مع الرزمة : إنتاج حالتين في حال الوصول إلى الهدف.

السيناريو الثاني: وجود عدة مسارات إلى الهدف وعدد عقد أكبر لكن مسار واحد يسمح للرزمة عبوره للهدف: إنتاج حالتين أيضاً.

بقية التجارب بسبب توافر عدة مسارات تتوافق قواعدها مع الرزمة وبالتالي يتم احتساب كافة الحالات التي تنتج عن البحث ضمن المسارات وهذا مايفسر زيادة عدد الحالات للوصول إلى النتيجة ولكنها مقبولة مقارنة بالنماذج الأخرى. تؤدي زيادة عدد العقد في الشبكة إلى زيادة عدد المسارات وكذلك عدد قواعد النفاذ المسندة للوصلات، وبالتالي فإن عملية تحديد الوصولية سيتطلب عدد حالات أكبر لأن أداة فحص النماذج ستقوم بتجريب كافة الطرق والقواعد وتحديد الحالات التي تحقق الفرضية للوصول إلى نهاية النمذجة، وفي حال عدم وجود أي حالة محققة تقوم بطباعة error trace ومتحولات التوقف.

تعالج الأدوات الأخرى عملية انتقال الرزمة على أنها من عقدة لعقدة وتغير الحالة سيكون عند كل انتقال وهذا مايفسر العدد الكبير من الحالات التي يتم الوصول إليها (تخفيض عدد الحالات يخفف من زمن استجابة النموذج)، أما في النموذج المقدم لدينا قمنا باختصار عدد الحالات بالطريقة التالية:

الحالة الابتدائية تدل على موقع الرزمة الحالي والإجراء المتبع للانتقال للحالات التالية هو انتقاء المسارات الموافقة لمصدر ومستقبل الرزمة ومن ثم عدد قواعد النفاذ،(يبقى النموذج في مكان الحالة الابتدائية في حال كان يختبر المسارات الغير متوافقه من ناحية القواعد) ويتم الانتقال إلى الحالة النهائية: التي هي إما أرسلت الرزمة أو لم ترسل عند انتهاء البحث ضمن كافة الاحتمالات.

العمل الأقرب لهذا العمل هو العمل المقدم في [8] لكنه لم يتميز بالمرونة بالإضافة الى عدم اختباره عملياً حيث تناول عملية تعداد حالات الشبكة التي تصلح لعملية وصولية الرزم (لكن لم يتم اختبار هذه النموذج على أدوات فحص النماذج) ، أما في هذا البحث قمنا بإضافة حالة العقد والوصلات من أجل حساب حالة الشبكة وتحديد الوصولية وتحديد خصائص أخرى، ومن أجل تخفيف الحالات الانتقالية التي تعاني منها هذه النماذج والتي تعتبر ناتجة عن معالجة ترويسات الرزم (فضاء كبير) [9] والانتقالات بين العقد، فإنه في هذا البحث تنخفض عدد الحالات بسبب اختبار فعالية المسار كاملاً وتوافقه مع القواعد المناسبة للرزمة بإجراء واحد، وبالتالي يعد هذه سبباً إضافياً لتخفيف عدد الحالات للحصول على النتيجة.

عملية تحديد الوصولية وفق هذا النموذج من الممكن أن تتم بشكل مسبق، وبالتالي يستطيع المدير التأكد من إمكانية اتصال عقدتين وفق بروتوكول معين في حال تعطل عقد أو مسارات معينة، وبالتالي يمكن تحديد العقد الهامة

والمسارات الحرجة بالنسبة لاتصال معين. كما يمكن استخدام هذا النموذج خلال عمل الشبكة بحيث يتم أخذ لقطات من الشبكة خلال فواصل زمنية معينة ويتم حساب الوصولية أو الخصائص المراد التأكد منها.

عملية اختبار النموذج تتم بطريقة ديناميكية أيضاً حيث يتلقى التوصيفات التي سيقوم باختبارها من خلال العلاقة (22) وبالتالي لايعتبر ستاتيكياً، وهذا يختلف عن النموذج الستاتيكي لتحليل قوائم النفاذ المقدم في [4] والذي يعد نموذجاً نظرياً كما يعد حلاً يتميز بالتعقيد . يبين الجدول (3) أهم النقاط التي تناولتها الأبحاث فيما يتعلق بموضوع البحث مقارنه بالحل المقدم.

الجدول (3) مزايا الحل المقدم مقارنة بالأبحاث ذات الصلة

	حالة الشبكة	مسارات التوجيه	تغير الطوبولوجيا	التضارب في القواعد	الوصولية	اختبار النموذج
Jefry[3]	x	x	*	*	*	*
Xie[4]	x	*	x	x	*	x
Bera[6]	*	x	x	*	x	*
Sveda [7]	*	*	x	x	*	x
Ryšavý [8]	*	*	*	x	*	x
Our approach	*	*	*	*	*	*

بالإضافة للمزايا السابقة فإنه يختلف عن أدوات فحص النماذج من خلال الإشارة للخطأ بطريقة فعالة أكثر من خلال المتحول h الذي يحتفظ بكل العقد التي مرت بها الرزمة وصولاً إلى العقدة التي تم التوقف عندها، وبالتالي تسهل على المدير تحديد الخطأ، وهذا مايختلف به عن أداة sat–solver المستخدمة في [3] و [6] التي تقوم بالإشارة إلى الأخطاء وتتوقف عند الحالة التي سببت الخطأ بدون معرفة الحالات السابقة التي أدت إلى هذا الخطأ.

الاستنتاجات والتوصيات:

تم في هذا البحث تطوير نموذج صوري لتوصيف الوصولية في الشبكات بشكل عام مع الأخذ بالحسبان قواعد النفاذ المعرفة في الشبكة والمسارات الموجودة بالإضافة لمراعاة حالة الشبكة التي تم تمثيلها منطقياً بأصفار وواحدات وذلك لتطوير مفهوم وصولية الرزم، حيث يتميز هذا النموذج عن غيره بديناميكيته وتجاوبه مع تغير الحالات للشبكة، إذ أن كتابته بصيغ رياضيه بلغة +TLA يمكن من اختبار العديد من المعايير في الشبكة سواء الوصولية أو التأكد من التعريفات واكتشاف التعريفات المتناقضة، وتوافق التوجيه مع المرشحات وهذا ما يميز التوصيف بلغة +TLA.

تم اختبار النموذج بداية عن طريق التحقق قواعدياً (Parsing) في الأداة ومن ثم اختباره لإعطاء نتائج على رزم معينة لمعرفة إذا ماكانت قابلة للوصول إلى الهدف أم لا واستخدام TLC التي تقوم بإنتاج الحالات المتوافقة مع ملف التوصيف وفق المدخلات المتعلقة بتعريفات الشبكة وطوبولوجيتها.

قدم النموذج المزايا الآتية:

- التحقق من الكثير من الخصائص وليس الوصولية فقط والتجاوب مع تغير الشبكة من خلال تغيير قيم المتحولات التي تشير إلى حالة العقد والوصلات.
- يصلح هذا النموذج للاختبار بشكل مسبق لتعريفات الشبكة ويمكن استخدامه خلال عمل الشبكة بأخذ لقطات من الشبكة خلال فواصل زمنية ومن ثم التحقق من الخصائص المطلوبة.
- مساعدة المدير على تحديد مكان الخطأ والحالات السابقة التي مرت بها الرزمة.
- تخفيض عدد الحالات للنظام الانتقالي بحيث يتم تخفيف مشكلة انفجار الحالات.

يعد هذا النموذج أساساً لتحليل قواعد النفاذ وخواص التسيير في الشبكات حيث يتميز بقابليته للتطوير وإدخال وحدات جديدة لمعالجة مشاكل أخرى في الشبكات، مثل تحديد مكان عملية سحب الرزم وعمليات مراقبة المسارات كثيرة الانقطاع، إضافة لمكاملته فعلياً مع الشبكات الحاسوبية بحيث يمكن كتابة ملف تنفيذي يقوم بتحويل العقد والعناوين والطوبولوجيا إلى شكل ملف التعريفات وبالتالي يصبح من الممكن استخدامه في الزمن الحقيقي، كما نعمل على استخدام النموذج السابق في عملية تحسين إدارة الشبكات في حال تعديل قواعد النفاذ أو خروج عقد هامة من الشبكة وذلك من خلال اختبار الوصولية في حال كانت العقد فعالة أو غير فعالة في مسارات معينة.

المراجع:

[1] - LAMSWEERDE,A.V., *Formal specification.*The conference on the future of Software engineering – USA,2000,147-159.

[2] -GUTTMAN., *Filtering postures: Local enforcement for global policies*".IEEE,1997,60-67.

[3]- JEFFREY,A., SAMAK,T., *Model checking firewall policy configurations. I*n: IEEE International Workshop on Policies for Distributed Systems and Networks, 2009, 60–67.

[4]-XIE,G., ZHAN,J., MALTZ,D., ZHANG,H., GREENBERG,A., HJALMTYSSO,G., REXFORD,J., *On static reachability analysis of ip networks*. IEEE infocom , 2005,Vol(3), 2170 – 2183.

[5]- BERA,P., GHOSH,S., DASGUPTA,P., *Formal analysis of security policy implementations in enterprise networks.* International Journal of Computer Networks and Communications, 2009,56–73.

[6]-BERA,P., GHOSH,S., DASGUPTA,P., *Formal Verification of Security Policy Implementations in Enterprise Networks.* In: Prakash, 5th International Conf ICISS. LNCS, vol. 5905, 2009,117-131.

[7]- SVEDA,M., RYSAVY,O., SILVA,G.,*Reachability Analysis in Dynamically Routed Networks.* 18th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, 2011, 197-205.

[8]- RYŠAVÝ,O.,DEG,S., MATOUŠEK,P., ŠVÉDA,M., *On formal reachability analysis in networks with dynamic behavior"*, In: Telecommunication Systems, Vol. 52, No. 2, 2013, New York, USA, ISSN 1018-4864,919-929.

[9]-KAZEMIAN,P., CHANG,M., ZENG,H., VARGHESE,G., MCKEOWN,N., WHYTE,S., *Real time network policy checking using header spaceanalysis.*In USENIX Symposium on Networked Systems Design and Implementation (NSDI),2013, 99-112.

[10]- BAIER,C. , KATOEN,J. , et al, *Principles of model checking*. Volume 26202649. MIT press Cambridge, 2008,994.

[11]- LAMPORT,L., *Specifying Systems*. Addison-Wesley Longman Publishing Co., Inc. version 18 June, 2002,382.

[12]-LAMPORT,L.,*The Temporal Logic of Actions*. ACM Transactions on Programming Languages and Systems,1994,872-923.

[13] -WANG,H., LIU,H., Guo,X., *Specify and Compose Web Services by TLA*. IEEE,2008, pp:766-767.

[14]-ZHANG,H., SONG,X., *Specifying time-sensitive systems with TLA*+.IEEE, 2013,pp: 425-430.

[15] - BALL, T., & BJØRNER, N., & GEMBER, A., & ITZHAKY, S., *VeriCon: Towards Verifying Controller Programs in Software-Defined Networks*. ACM,2014, pp: 1-12.

[16] - SETHI, D., & NARAYANA, S., & MALIK, S., *Abstractions for Model Checking (SDN) Controllers.* IN FMCAD .2013.pp:1-8.