دراسة وتطوير نموذج شبكة هجينة مُعرَّفة برمجياً لتحسين أداء الشَّبكة التَّقليديَّة

م. محمد الشَّمري*
و د. علي حمَّاد**
و د. خلدون خُرّزم***

المُلَخّص

تُساعِد الشَّبكات الهجينة المُعرَّفة برمجياً على الانتقال من الشَّبكات التَّقليديَّة إلى الشَّبكات المُعرَّفة برمجياً إذ تُعتبَر مرحلة انتقاليَّة ضروريَّة وحتميَّة. وذلك بهدف الوصول إلى شبكات مُعرَّفة برمجيَّاً لما تُقدِمَه من خدمات مثل قابليَّة البرمجة والتَّحكم المركزي والذي تفتقر إليه الشَّبكات التَّقليديَّة.

يُحقِق نموذج الشَّبكة الهجينة المُعتَمد، التَّواصل بين جزء الشَّبكة التَّقليديَّة وجزء الشَّبكة المُعرَّف برمجياً، وذلك عن طريق إضافة مُتحكِّم هجين يعمل على تنسيق العمل بين الأجزاء المختلفة للشَّبكة الهجينة. كما يُساعد ذلك النموذج المُقترح على تحسين أداء بعض وظائف الشَّبكة التَّقليديَّة.

نسعى في هذا العمل على تحسين زمن تقارب بروتوكول التَّوجيه OSPF بين المُوجهات التَّقليديَّة وذلك بالاعتماد على نموذج الشَّبكة الهجينة المُقترح، حيث نقوم بالمُقارنة بين زمن تقارب OSPF في الشَّبكة التَّقليديَّة وفي الشَّبكة الهجينة. إذ نحصل على نتائج المُقارنة بعد إجراء التَّجارب على عدد من طوبولوجيا الشَّبكات المُختلفة الحجم.

كلمات مفتاحية: الشَّبكات المُعرَّفة برمجياً، نموذج شبكة هجينة مُعرَّفة برمجياً، زمن تقارب OSPF.

*طالب ماجستير في قسم النّظم والشَّبكات الحاسوبيَّة، كليَّة الهندسة المعلوماتيَّة، جامعة دمشق، دمشق سوريا.
**مُدرّس في قسّم النّظم والشَّبكات الحاسوبيَّة، كلية المعلوماتيَّة، جامعة دمشق، دمشق سوريا.
***أستاذ في قسّم الاتِّصالات، المعهد العالي للعلوم التطبيقيَّة والتكنولوجيا، دمشق سوريا.

Study and deployment of Hybrid SDN model for enhancing legacy network performance

Eng. Mohammed Al-Shammari [*]
Dr. Ali Hammad[**]
Dr. Khaldoun Khorzom[***]

Abstract

Hybrid-SDN networks are an inevitable and necessary step to achieve the transition from the legacy networks to SDN networks. That is to benefit from all the services that SDN offers such as programmable and central control.

By adding a hybrid controller, the proposed hybrid-SDN model provides connectivity between traditional and the SDN parts. It also enhances the performance of some traditional network functionalities.

The paper aims to improve the OSPF convergence time for legacy routers. We used the proposed hybrid-SDN model to compare the OSPF convergence time between legacy networks and hybrid-SDN networks for multiple topologies with different sizes.

Key words: software defined networks, Hybrid SDN, OSPF Convergence time.

[*] Master student, Department of Systems and Computer Networks, Information Engineer Faculty, Damascus university, Damascus Syria

[**] Lecturer, Department of Systems and Computer Networks, Information Engineer Faculty, Damascus university, Damascus Syria.

[***] Professor, Department of Telecommunication, High Institute for Application Science and Technology, Damascus Syria.

المُقدِّمة

تعتمد البُنية التَّحتيَّة للشَّبكات الحاليَّة على الشَّبكات التَّقليديَّة، لذلك يحتاج الانتقال بهذه الشَّبكات نحو الشَّبكات المُعرَّفة برمجياً لتغيير كامل أجهزة الشَّبكة الموجودة حالياً أو تحديث أنظمة تشغيلها لتصبح تدعم برتوكولات الشَّبكات المُعرَّفة برمجياً (مثل بروتكول OpenFlow المسؤول عن التَّواصل بين طبقة التَّحكم وطبقة البيانات)، وغالباً لن يكون ذلك مُمكن بسبب طبيعة تكوين أجهزة الشَّبكة التَّقليديَّة الّتي تكون في معظم الأحيان مُحدَّدة الوظيفة وأنظمة تشغيلها ذات ارتباط وثيق بالشَّركة المُنتجة فهي غالباً ما تكون الجهة الوحيدة التي تطوّر أجهزتها ومسؤولة عن تحديث أنظمتها. ندرس في هذا العمل نموذج شبكة هجينة قابل للتطبيق، يساعد على عملية الانتقال من الشَّبكات التَّقليديَّة إلى الشَّبكات المُعرَّفة برمجياً. يسمح ذلك النَّموذج الهجين بتواجد أجهزة الشَّبكة التَّقليديَّة جنباً إلى جنب مع أجهزة شبكة SDN (Software Defined Network). كما أنه يُحقق التَّواصل والتَّكامل بين وظائف طبقة التَّحكم الموزعة ضمن الأجهزة التَّقليديَّة وطبقة التَّحكم المركزيَّة في شبكة SDN.

يُمكن تحقيق العديد من المكاسب بالمكاملة بين وظائف طبقات التحكم التَّقليديَّة وSDN. إذ إنَّ وجود مُتحكِّم مركزي بالشَّبكة الهجينة يعطي القدرة على التَّحكم بعدد من وظائف الأجهزة التَّقليديَّة بشكل مشابه لآلية التَّحكم المركزيَّة ضمن شبكة SDN، مما يساعد على تحسين عمل بعض تلك الوظائف.

نسعى في هذا العمل على تحسين زمن تقارب بروتوكول التَّوجيه OSPF ضمن الشَّبكة الهجينة وذلك بالاعتماد على النَّموذج الذي تمَّ طرحه، مع العلم أن بروتوكول التَّوجيه OSPF تمَّ الاعتماد عليه ضمن نموذج الشَّبكة الهجينة المُقترح لتحقيق التَّواصل بين جزء الشَّبكة التَّقليدي وجزء الشَّبكة المُعرف برمجياً. يحوي نموذج الشَّبكة الهجينة مُتحكِّم هجين يتكون من عنصرين أساسيين الأول هو مُتحكِّم SDN (مفتوح المصدر) يدعى POX للتحكم بمبدلات OF (OpenFlow) والثَّاني هو عبارة عن مُخدم توجيه تقليدي (Legacy Route Server) يقوم بتشغيل برمجيَّة تُدعى Quagga، يعمل ذلك المُخدم كموجه تقليدي يشغّل بروتوكول التَّوجيه OSPF ويلعب دوراً هاماً بتخزين معلومات طوبولوجيا الشَّبكة التَّقليديَّة.

نناقش في القسم التَّالي من هذا العمل الأعمال السَّابقة ضمن مجال الشَّبكات الهجينة Hybrid SDNs، أما في القسم الثَّاني من هذا العمل سنناقش نموذج الشَّبكة الهجينة الذي سيتمُّ إجراء الاختبارات عليه، ونقوم بتوضيح كيفيَّة عمل ذلك النَّموذج ضمن القسم الثَّالث. ونعرض بالقسم الرَّابع التَّجارب الّتي تمَّ إجرائها والنَّتائج الّتي تمَّ الحصول عليها. وأخيراً ينتهي العمل بالخاتمة والّتي نعرض ضمنها الأعمال المستقبليَّة المُمكنة في مجال هذا العمل.

I. الأعمال السَّابقة

نعرض فيما يلي بعض الأعمال الأكثر شيوعاً ضمن مجال الشَّبكات الهجينة المُعرَّفة برمجياً هي: RouteFlow [11] و LegacyFlow [5] و Panopticon [8] و Cardiagn [12] و HybNET [9] و Symphony [2] وأيضاً I2RS [4].

في RouteFlow يتمُّ محاكاة جميع مُبدلات OF بشكل افتراضي كأجهزة طبقة ثالثة وهي تقوم

بالرَّبط بين نطاق الشَّبكة التَّقليديَّة وشبكة SDN. ولكن سلبيات هذا النِّظام هو أنه يزيد من تعقيد العمل ويصبح هناك العديد من نقاط الضَّعف ضمن الشَّبكة كما أنَّ توقُّف مُخدّم RouteFlow (RF) يؤدِّي للعزل التَّام لنطاق شبكة SDN. في I2RS يتمُّ استخدام ForCES [14] كبروتوكول للتَّواصل بين طبقة التَّحكُّم وطبقة البيانات، التَّحدِّي الرَّئيسي في هذا العمل هو التَّأقلم والتَّكيف بين بروتوكول ForCES وباقي بروتوكولات التَّواصل. يُعتَبر Cardigan هو تنجيز لـ RouteFlow لذلك فهو يُعاني من نفس التَّحديات الَّتي تواجه RouteFlow.

يقوم Panopticon بتحقيق التَّكامل بين شبكة SDN والشَّبكة التَّقليديَّة عند الطَّبقة الثَّانية (Data link) باستخدام تقنيَّة VLANS لكن تطبيق السِّياسة الشَّبكيَّة، على سبيل المِثال ACLs لايزال يتم عن طريق إعداد كل جهاز شبكي على حدة. بالنِّسبة للعمل LegacyFlow يُعالج حالياً التَّحديات ضمن الطَّبقة الثَّانية، حيث أنه من غير الواضح فيما إذا كانت الأجهزة التَّقليديَّة الأخرى ضمن الطَّبقة الثَّالثة مثل المُوجِّهات هل سيتمُّ إعدادها من قبل نظام LegacyFlow أم لا. HybNET يقوم بالتَّكامل بين وظائف كلا النَّوعين من الشَّبكات ضمن الطَّبقة الثَّانية ولكن عدد الشَّبكات الافتراضيَّة (VLANs) الَّتي يُمكن إنشاؤها محدودة بالعدد 4096 وذلك ينعكس كقيد على عدد الشَّرائح الَّتي يُمكن استخدمها ضمن الشَّبكة الهجينة. ضمن نموذج Symphony يتم اقتراح نموذج شبكة هجينة يعتمد على مُتحكِّم هجين يُحقق التَّواصل بين الشَّبكة التَّقليديَّة وشبكة SDN إلا أنَّه لا يتم توضيح آلية عمل النَّموذج المُقترح ولا حتى كيف يُمكن الاعتماد على ذلك النَّموذج لتحسين أداء وظائف الشَّبكة التَّقليديَّة.

من ناحية أخرى هناك عمل يُدعى SUMA [3] يؤمِّن وسيلة مُراقبة مُوحَّدة بين نظام الإدارة التَّقليدي والمُتحكِّم في شبكة SDN ومُبدلات OF. لكنه لا يقترح إي وسيلة للتواصل بين كلا النطاقين التَّقليدي والمُعرَّف برمجياً. هناك أعمال أخرى صدرت مؤخراً مثل العمل ضمن [1] والذي يُساعد على تحسين وجود الشَّبكة التَّقليديَّة وذلك عن طريق تقسيمها لعدت نطاقات وإنشاء اتصال بين نطاقات OSPF عن طريق التَّحكُّم بمبدلات OF.

بينما نقترح في هذه العمل استخدام مُتحكِّم هجين يتكون من مُتحكِّم SDN ومُخدم توجيه تقليدي لتشكيل علاقات تجاور مع المُوجهات التَّقليديَّة باستخدام بروتوكول التَّوجيه OSPF وذلك بالاعتماد على النَّموذج المُقترح ضمن [2] وكذلك العمل [7] الذي يوضّح آلية التَّواصل بين أجزاء الشَّبكة الهجينة. ذلك النَّموذج يسمح بإجراء اختبارات نبين من خلالها أنه لا يتأثر زمن تقارب برتوكول التَّوجيه OSPF بعدد المُبدلات في الشَّبكة الهجينة كما هو الحال في الشَّبكة التَّقليديَّة، بينما يتم التَّركيز في كافة الأعمال المذكورة سابقاً على بنية الشَّبكة الهجينة دون ذكر القيمة المُضافة لتلك البُنية على أداء وظائف الشَّبكة التَّقليديَّة.

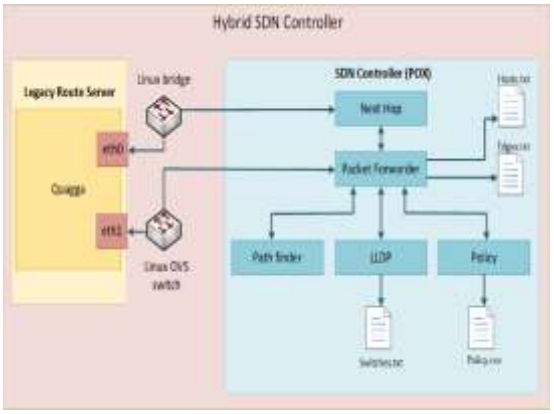
II. نموذج الشَّبكة الهجينة المُقترح

يهدف المُتحكِّم الهجين المُقترح إلى تحقيق التَّواصل بين أجزاء الشَّبكة الهجينة (تقليديَّة وSDN)، وذلك من خلال ربط طبقة تحكم SDN مع الأجهزة التَّقليديَّة. سنُقدم في هذا القسم بُنية المُتحكِّم الهجين وعناصره بشيء من التَّفصيل.

واحد من أهم العوامل الدَّافعة للتصميم هو السَّماح للمُتحكِّم بتثبيت قواعد التَّدفُّق بشكل

استباقي ضمن طبقة البيانات تجنباً للتأخر النّاجم عن حساب قواعد التَّدفُّق كرد فعل على وصول طرود جديدة ومن ثم تثبيت قواعد التَّدفُّق. وبالتالي يتكوَّن المُتحكِّم الهجين من مكونين أساسيين، مُخدّم التَّوجيه التَّقليدي Legacy Routing Server (LRS) ومكوّن توجيه الطُّرود Packet-Forwarder بهدف تحديد المسار بين المصدر والهدف ومن ثمَّ تثبيت قواعد التَّدفُّق ضمن مُبدلات OF الَّتي تُشكّل المسار.

توجد المُكوِّنات Packet-Forwarder و Path–finder و LLDP و Next–hop ضمن مُتحكِّم SDN، أما بالنسبة لـ LRS يُعتبر مخزن للمسارات التَّقليديَّة. يُمكن اعتبار أنَّ LRS و Packet-Forwarder جنباً إلى جنب يُشكلان المُتحكِّم الهجين. هناك مكونات غير نشطة بالإضافة للمكوّنات السَّابقة وهي عبارة عن ملفات مثل Hosts.txt و Switches.txt و Edges.txt من أجل تخزين معلومات طوبولوجيا الشَّبكة والَّتي يتمُّ استخدامها من قبل المُتحكِّم لتثبيت قواعد التَّدفُّق بشكل استباقي ضمن مُبدلات OF المُحددة لتشكيل المسار المطلوب. نوضّح بُنية المُتحكِّم الهجين ضمن الشَّكل 1. بحيث نعتبر مُبدل OF أنه عقدة هدف في حال كان مُتصل مع المُستخدم الهدف مُباشرة أو يكون مُبدل حدودي في حال كان مُتصل مع مُوجِّه تقليدي. أما مُبدلات OF الَّتي تشكل المسار تُسمّى المُبدلات الوسيطة. نتحدث فيما يلي عن مكونات المُتحكِّم الهجين بشيء من التفصيل.



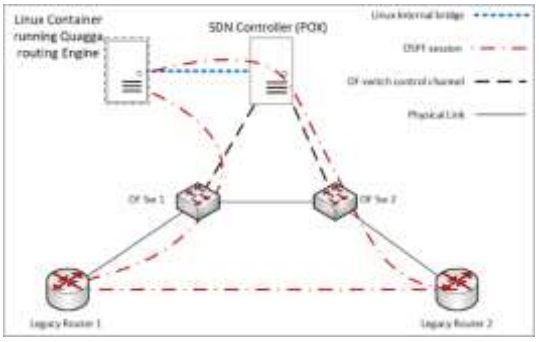
الشكل 1 بنية المُتحكِّم الهجين

1. مُكوّن توجيه الطُّرود Packet-Forwarder:

يقوم مُكوّن توجيه الطُّرود بالوظائف الأساسيَّة للمُتحكِّم، من خلال التَّنصُّت والتقاط أحداث OF الَّتي تنتج عن طبقة البيانات. مُكوّن توجيه الطُّرود هو عبارة عن سلسلة من حالات اتخاذ القرار الَّتي ينفذها المُتحكِّم عندما يستقبل أحداث OF. يتم التقاط أحداث OF في هذا النموذج من نوع ConnectionUp, PacketIn, PortStatus.

2. مخدم التَّوجيه التَّقليدي Legacy Routing Server (LRS):

وهو عبارة عن نظام Linux يقوم بتشغيل برمجيَّة مُحرك التَّوجيه Quagga [10]. فمن خلال إنشاء علاقة تجاور OSPF مع الشَّبكة التَّقليديَّة، يتشكل لدى LRS نظرة منطقية شاملة لكامل الشَّبكة التَّقليديَّة. يقوم المُتحكِّم باستشارة LRS لاختيار أفضل عقدة تالية للوصول إلى الهدف البعيد الذي يقع ضمن الشَّبكة التَّقليديَّة. نوضّح بالشَّكل 2 علاقة التَّجاور بين LRS والمُوجهات الحدوديَّة.



الشكل 2 التَّواصل بين LRS والمُوجهات الحدوديَّة

3. مُكوّن تحديد المسار Path-finder:

يُساعد هذا المُكوّن المُتحكِّم على إيجاد المسارات المثاليَّة ضمن جزء شبكة SDN من خلال تنفيذ خوارزميَّة Dijikstra's لتحديد أقصر مسار وذلك باستخدام ملف switches.txt كدخل للخوارزمية. حيث أنَّ switches.txt هو عبارة عن بُنية الشَّبكة والذي تمَّ تشكيله من قبل المُكوّن LLDP. يحدد مُكوّن تحديد المسار مجموعة من مُبدلات OF الَّتي تُشكل ذلك المسار بين المصدر والهدف ويعمل المُتحكِّم على تثبيت قواعد التَّدفُّق ضمن تلك المُبدلات.

4. مُكوّن العقدة التَّالية Next-hop:

يتواصل هذا المُكوّن مع مُخدّم التَّوجيه التَّقليديَّ (بمساعدة OVS ضمن بيئة Linux وما يُقدمه LRS من تسهيلات للحصول على جدول التَّوجيه لديه) ويستفسر عن أفضل مُوجِّه حدودي للوصول إلى الهدف البعيد. بعدها يُعيّن مُكوّن توجيه الطُّرود مُبدل OF الموافق والمُتصل مع المُوجه التَّقليدي الحدودي كمُبدل OF هدف، يتبع ذلك تحديد مجموعة من مُبدلات OF الوسيطة بين المصدر والهدف لتوجيه الطُّرود عبرها. يُثبت المُتحكِّم بعد ذلك قواعد التَّدفُّق المُناسبة ضمن جداول مُبدلات OF الَّتي تُشكل المسار.

تساعد المُكوِّنات الأخرى الغير نشطة (مثل ملفات Host و Switches و Edges) المُتحكِّم على حساب وتثبيت قواعد التَّدفُّق المُناسبة بشكل استباقي ضمن مُبدلات OF. في حال عدم احتواء تلك الملفات على معلومات، يُحدد المُتحكِّم قواعد التَّدفق ويُثبتها عند لحظة التقاط الأحداث، يُمكن أن يُسبب ذلك تأخير في الشَّبكة. لذلك فإن حفظ معلومات طوبولوجيا الشَّبكة ضمن ملفات مُعيَّنة يجعل المعلومات مُتوفِّرة بسهولة للمُتحكِّم، ويصبح قادراً على تثبيت قواعد التَّدفُّق بشكل استباقي مما يوفر الوقت ضمن الشَّبكة أثناء العمل.

III. آلية العمل

يلتقط المُتحكِّم الهجين رسائل OF من نوع مُحدّد فقط وهي PacketIn و ConnectionUp و PortStatus. بناء على التقاط حدث ConnectionUp يعمل مُكوّن LLDP ضمن المُتحكِّم على اكتشاف مُبدلات OF ويُحدِّث المعلومات ضمن ملف Switches بما يوافق ذلك.

يعمل مُكوّن مُوجِّه الطُّرود (Packet-Forwarder) بسلوك مشابه لسلوك مُوجِّه تقليدي (ضمن الطبقة الثَّالثة) بالإضافة لالتقاط أحداث OF من طبقة البيانات. يُنفذ هذا المُكوّن وظائف مثل unicast IP و multicast IP وذلك لالتقاط الطُّرود التَّقليديَّة سواء أكانت unicast أم multicast، كما يقوم أيضاً بأفعال تتعلق بعمل برتوكول OF مثل ترتيب الطُّرود ضمن الأرتال في طبقة البيانات، وتثبيت أو إزالة قواعد التَّدفُّق وذلك باستخدام رسائل OF مثل ofp_flow_mod أو ofp_packet_out.

يعالج المُتحكِّم الطُّرود الواردة من خلال التَّحقُّق من حمل الطَّرد (payload) ومن ثمَّ يُحدد مسار العمل للتعامل مع تلك الطُّرود.

على سبيل المثال، إذا كان حمل الطُّرود الواردة من نوع routing control multicast (OSPF hello)، يُحدد المُتحكِّم مجموعة من مُبدلات OF كهدف (عادة تكون هي مُبدلات OF الحدودية) لوضع الطُّرود ضمن الأرتال الخاصَّة بتلك المُبدلات، إلى جانب ذلك، يضع المُتحكِّم الطُّرود ضمن رتل OVS المتصل مع مُكوّن LRS باعتباره مُبدل حدودي.

وبالتالي يستقبل LRS تحديثات OSPF من الشَّبكة التَّقليديَّة ويُنشِئ علاقات تجاور مع المُوجِّهات الحدودية. في الحالات الَّتي يكون فيها الطَّرد مُتجه نحو هدف محدَّد، يعتمد المُتحكِّم على مُكوّن unicast، مما يستدعي عملية اكتشاف المسار بين المصدر والهدف، يتبع ذلك تَّثبيت قواعد التَّدفُّق بشكل استباقي على كافة مُبدلات OF الوسيطة الَّتي تشكل المسار.

يستقبل مُحرك التَّوجيه Quagga الذي يعمل ضمن LRS تحديثات بروتوكول التَّوجيه OSPF (من خلال المنفذ eth1 ضمن LRS المُوضَّحة بالشَّكل 1) لتتشكل علاقات تجاور OSPF مع المُوجِّهات التَّقليديَّة الحدودية كافةً ويتمُّ بناء طوبولوجيا الشَّبكة التَّقليديَّة. أخيراً، يتمُّ تخزين تفاصيل مُبدلات OF الحدوديَّة والمُوجِّهات التَّقليديَّة الحدوديَّة الموافقة والمتَّصلة بتلك المُبدلات في الملف Edges.txt. يُمكن اعتبار الوظيفة السَّابقة هي إنشاء علاقات تجاور OSPF عبر الشَّبكة المحليَّة LAN.

1. التَّواصل بين المُتحكِّم POX ومخدم التَّوجيه التَّقليدي LRS

يعمل مُتحكِّم SDN والذي يُدعى POX [6] على نفس المخدم مع LRS الذي يعمل ضمن container Linux (كما هو موضَّح بالشكل 1). يتواصل المُتحكِّم POX مع مُخدّم التَّوجيه التَّقليدي LRS عبر مُبدل OF الموجود بشكل أساسي ضمن بيئة (OpenVSwitch OVS) Linux من خلال المنفذ eth1 كما يتمُّ إدارة OVS من قبل المُتحكِّم POX، ذلك يسمح للمُتحكِّم بالتفاعل مع LRS كأي طرفية مُتصلة مع مُبدل OF. كما أن المنفذ eth0 ضمن LRS يتصل مع المُخدم المضيف من خلال Linux Bridge، وبما أن المُخدم الذي يحوي المُتحكِّم POX ومُخدم التَّوجيه التَّقليدي LRS يتَّصل فيزيائياً بالشَّبكة من خلال أحد مُبدلات OF، يتمُّ التَّعامل مع LRS من خلال المُكوّن Next-hop الموجود ضمن المُتحكِّم وذلك من خلال المنفذ eth0. ذلك يسمح للمُتحكِّم POX بالحصول على معلومات الشَّبكة التقليديَّة الموجودة ضمن LRS بمساعدة المُكوّن Next-hop المُتصل مع LRS عن طريق eth0، تتم تلك العمليَّة بالاستفادة من البُنية البرمجيَّة لـ Quagga والَّتي يُمكن الحصول منها على جدول توجيه الشَّبكة التَّقليديَّة.

بما أن LRS لديه معلومات الشَّبكات التَّقليديَّة البعيدة فهو المسؤول عن إيصال الطُّرود إليها بالتَّنسيق مع مُكوّن Next-hop.

2. آليَّة التقاط البيانات

يقوم المُتحكِّم بثبيت قواعد التَّدفُّق على مجموعة من مُبدلات OF الَّتي تصل إلى الهدف وذلك بناء على استقبال الطَّرد. وبالاعتماد على عنوان الهدف للطَّرد يختار المُتحكِّم مُبدل OF الهدف ومجموعة من مُبدلات OF الوسيطة (الَّتي تشكل المسار). مثلاً إذا كان الطَّرد من نوع unicast، يكون المُبدل الهدف هو عبارة عن مُبدل OF وحيد. ومُبدلات OF الَّتي تشكل

المسار بين مُبدل OF المصدر ومُبدل OF الهدف يتمُّ تعيينها كمُبدلات OF وسيطة. وبمساعدة الملف Hosts.txt، يتمَّ تحديد فيما إذا كان عنوان الهدف خاص بعقدة مُتصلة مع مُبدل OF (أي تقع ضمن جزء شبكة SDN)، عندها يتم تعيين ذلك المُبدل كمُبدل OF هدف ويتم حاسب المسار وتحديده. في حال العُقدة الهدف لم تكن ضمن مجال شبكة SDN، يبحث المُتحكِّم ضمن ملف Edges.txt لتحديد فيما إذا كان هدف الطَّرد هو أحد المُوجِّهات الحدوديَّة. عندها يُعيّن المُتحكِّم مُبدل OF الحدودي المُتصل بذلك المُوجه كمُبدل OF هدف. في حال لم تكن العُقدة الهدف مُتصلة بمُدل OF ولا مُتصلة أيضاً بموجه حدودي عندها يتواصل المُتحكِّم POX مع مُكوّن LRS لتحديد فيما إذا كان العنوان يقع ضمن الشَّبكة التَّقليديَّة، في هذه الحالة يتمُّ تعيين المُوجه الحدودي (العُقدة التَّالية Next-hop) المُناسب وتعيين مُبدل OF المُتصل به (مُبدل OF الهدف). بعد ذلك يتمُّ تحديد المسار الأفضل بين مُبدل OF المصدر ومبدل OF الهدف ويثبت المُتحكِّم قواعد التَّدفُّق بشكل استباقي على مُبدلات OF المُحدِّدَة للمسار.

بشكل مُشابه، إذا كان الطَّرد مُتوجه نحو عدّة أهداف (multicast)، يتم اختيار مجموعة من مُبدلات OF هدف. على سبيل المِثال، في حال كان الطَّرد من نوع multicast OSPF، يتمُّ تعيين مُبدلات OF الحدوديّة كمُبدلات هدف. أمّا في حال طرود من نوع broadcast L2، تم تعيين منافذ جميع المُبدلات المُتصل بها عُقد طرفيَّة لتمرير الطُّرود عبرها.

يقوم المُتحكِّم POX بجميع الخطوات السَّابقة من خلال ضبط الإعدادات الّتي تحدد آليَّة عمل ذلك المُتحكِّم والّتي تكون ضمن ملفات مُخصصة [6].

يعمل ملف Edges.txt كمخزن للمُتحكِّم POX حيث يقوم بالبحث فيه قبل التَّواصل مع مُكوّن LRS. هذه الميزة مفيدة لمعالجة الطُّرود المُحدود عملها بالزمن مثل الطُّرود الَّتي تعمل على إنشاء علاقات تجاور OSPF. يتواصل المُتحكِّم مع مُكوّن LRS لإيجاد العُقدة التَّالية المُمكنة للوصول إلى الهدف. يحصل المُتحكِّم بعد ذلك على تفاصيل معلومات المُوجه الذي يُمثل العُقدة التَّالية من LRS ويحدد مُبدل OF الحدودي المُتصل بذلك المُوجه من خلال ملف Edges.txt.

تجدر الإشارة إلى أنَّ المُتحكِّم لا يقوم بأي حالة ببث الطُّرود على كامل الشَّبكة (broadcast). مثلاً لا يتمُّ تمرير الطُّرود من نوع (ARP) إلى المنافذ المُتَّصلة مع المُبدلات الأخرى، ولكن فقط يتم تمريرها إلى المنافذ المُتَّصلة مع الطَّرفيات الشَّبكيَّة النهائية. تمنع عملية التَّحقُّق تلك حدوث حلقات الطَّبقة الثَّانية أو ما يسمى broadcast storm ضمن الشَّبكة. علاوة على ذلك، في حال كان الطَّرد مُتجه نحو الشَّبكة التَّقليديَّة، يُعدل المُتحكِّم ترويسة ذلك الطرد ليصل إلى المُوجه الحدودي المُناسب. ذلك يضمن أنَّ المسار الذي تمَّ اختياره متاح وجميع وصلاته فعَّالة وأنه أفضل مسار ضمن الشَّبكة.

IV. الاختبارات والنتائج

نوضّح في هذا القسم من العمل حالة الاختبار الَّتي تمَّ تصميمُها بهدف تقييم نموذج الشَّبكة الهجينة المُقترح. حيث تم قياس الزَّمن اللازم لإنشاء علاقات التَّجاور بين المُوجِّهات الَّتي

تُشغّل بروتوكول OSPF، ودراسة تأثير عدد المُبدلات على زمن التَّقارب في كل من الشَّبكة التَّقليديَّة ونموذج الشَّبكة الهجينة المُقتَرح والمقارنة بينها.

نتحقق من الزَّمن الذي تستغرقه عقد الشَّبكة الطرفيَّة (المُوجِّهات الحدوديَّة) لتشكيل علاقات التَّجاور الخاصة ببروتوكول التَّوجيه OSPF وذلك عبر نطاق شبكة SDN. يتمّ تشغيل بروتكول Spanning Tree Protocol (STP) في الشَّبكات التَّقليديَّة من قِبل مُبدلات الطَّبقة الثَّانية (Data link) لتفادي مُشكلة الحلقات ضمن الشَّبكة. يحول بروتوكول STP الشَّبكة الحلقيَّة إلى شبكة شجريَّة وذلك عن طريق وضع بعض منافذ المُبدلات بوضعيَّة التمرير وبعضها الآخر في حالة التوقُّف عن العمل. تلك العمليَّة لا تقوم فقط بإنهاء الحلقات ولكن من المُمكن أنَّ تجعل المسارات طويلة بين عقد الشَّبكة. لذلك يستخدم المُتحكِّم الهجين بروتوكول LLDP لاكتشاف طوبولوجيا شبكة SDN وتخزين معلوماتها ضمن قاعدَّة بيانات الطوبولوجيا مما يقلِّل الاعتماد على بروتوكول STP بهدف الوصول إلى شبكة خالية من الحلقات.

1. زمن تقارب بروتوكول STP

يعمل بروتكول STP في الطَّبقة الثَّانية (Data link layer) لتشكيل طوبولوجيا خالية من الحلقات (شجريَّة على وجه التحديد) انطلاقاً من طوبولوجيا حلقيَّة. يبدأ بروتوكول STP باختيار مُبدل جذر (Root Bridge) للشَّبكة يتبع ذلك وضع بوابات المُبدلات في إحدى الحالات التالية (blocking, listening, learning, forwarding, disabled). بعد المرور بحالة listening وحالة learning، يتمُّ نقل البوابة لحالة forwarding. المجموع الكلي للوقت الذي تستغرقه كل خطوة يصل تقريباً لـ 45 ثانية. لذلك، تمَّ تنفيذ عدَّة إصدارات من بروتوكول STP، منها على سبيل المثال بروتوكول Rapid Spanning Tree Protocol (RSTP) وذلك للتَّخفيف من طول فترة زمن التَّقارب الَّتي يستغرقها بروتوكول STP. يعتمد أيضاً زمن التَّقارب لبروتوكول STP على عدد المُبدلات ضمن الشَّبكة. إذ إنه كلما زاد عدد مبدلات الشَّبكة يزداد تأخير زمن التَّقارب يُمكن حساب ذلك الزمن بشكل نظري اعتماداً على ما هو مطروح ضمن [13] وفق الصّيغة الرّياضية التَّالية:

$$\text{STP convergence time} = \frac{BP + MA + MF + MT}{2}$$

حيث:

BP: end-to-end BPDU_Propagation_delay

MA: Message_age_overestimate

MF: Maximum_farme_lifetime

MT: Maximum_transmission_halt_delay

نجد ضمن [13] توصيف للصّيغة السَّابقة والَّتي تعطينا الزَّمن اللازم لتقارب برتوكول STP بناءً على عدد المُبدلات ضمن الشَّبكة التَّقليديَّة دون التَّطرُق لتفاصيل وطبيعة طوبولوجيا الشَّبكة. وبالاعتماد على تلك الصّيغة الرّياضيَّة، يمكن استنتاج زمن تقارب بروتوكول STP انطلاقاً من عدد المبدلات التَّقليديَّة ضمن الشَّبكة كما هو موضَّح بالجَّدول رقم 1:

الجدول 1 تغير زمن تقارب بروتوكول STP بتغير عدد المُبدلات

عدد المُبدلات التَّقليديَّة	BP (ثا)	MA (ثا)	MF (ثا)	MT (ثا)	الزَّمن التقارب (ثا)
3	10	2	3.5	1	**8**
5	12	4	5.5	1	**11**
7	14	6	7.5	1	**14**
14	21	13	14.5	1	**25**
21	28	20	21.5	1	**35**
28	35	27	28.5	1	**46**

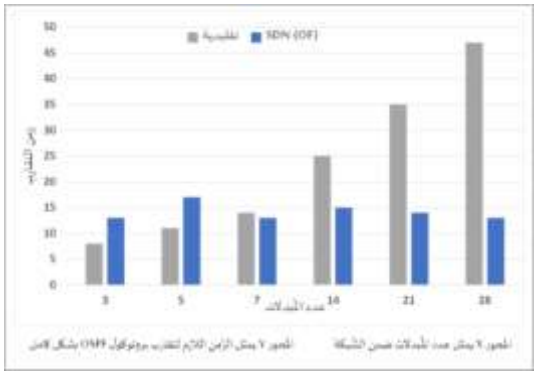
2. مقارنة زمن تقارب برتوكول OSPF عبر الشَّبكة التَّقليديَّة والشَّبكة الهجينة

يتمُّ تبادل طرود hello OSPF بعد أنَّ تكتمل عملية تقارب برتوكول STP. لذلك فإنه من الطَّبيعي افتراض أنَّ الزَّمن الذي تستغرقه المُوجِّهات التَّقليديَّة للوصول إلى حالة التَّقارب الكاملة هو أكبر من زمن تقارب بروتوكول STP. نوضّح بالجَّدول التَّالي رقم 2 مقارنة بين زمن تقارب بروتوكول OSPF عبر الشَّبكة التَّقليديَّة وعبر الشَّبكة الهجينة:

الجدول 2 مقارنة زمن تقارب برتوكول OSPF بين الشَّبكة التَّقليديَّة والشَّبكة الهجينة

عدد المُبدلات OF بالشبكة الهجينة	زمن التَّقارب عبر الشَّبكة التَّقليديَّة	زمن التَّقارب عبر الشَّبكة الهجينة
3	**>8**	**13**
5	**>11**	**17**
7	**>14**	**13**
14	**>25**	**15**
21	**>35**	**14**
28	**>46**	**12**

يشير العمود الأخير ذو العنوان (زمن التَّقارب عبر الشَّبكة الهجينة) إلى الزَّمن الذي تستغرقه المُوجِّهات التَّقليديَّة لتشكيل علاقات تجاور كاملة عبر شبكة SDN ابتداء من اللحظة الَّتي يبدأ فيه المُتحكِّم بالعمل. بهدف قياس الزَّمن بشكل دقيق تمَّ استخدام وظيفة time() في لغة برمجة python ، كما تمَّ أيضاً استخدام وظائف تحقُّق (debug functions) على المُوجِّهات لتحديد لحظة استقبال طرود OSPF بدقة. أما العمود الثَّاني (زمن التَّقارب عبر الشَّبكات التَّقليديَّة) يشير إلى الزَّمن اللازم للوصول إلى حالة التَّجاور الكاملة عبر المُوجِّهات التَّقليديَّة. يمكن أنَّ نستخلص مما سبق، أنَّه لا يوجد اختلاف كبير بالزَّمن اللازم للتقارب مهما اختلف عدد مُبدلات OF ضمن شَّبكة SDN في نموذج الشَّبكة الهجينة، بينما في حالة الشَّبكة التَّقليديَّة، يزداد الزَّمن اللازم لتحقيق تقارب STP كلما ازداد عدد المُبدلات (التَّقليديَّة)، مما يؤدي لازدياد الزَّمن الكلي اللازم لإنهاء عملية التجاور بين الموجهات التَّقليديَّة التي تُشغّل بروتوكول التَّوجيه OSPF. لذلك، يُمكن استنتاج أنَّ المُوجِّهات، عندما تكون مُتصلة فيما بينها عبر مُبدلات تقليديَّة (تعمل ضمن طبقة Data Link)، تحتاج لوقت كبير للوصول إلى حالة تقارب OSPF مُقارنة مع المُوجِّهات الَّتي تتصل عبر مُبدلات OF. لذلك فإنَّ المُتحكِّم الهجين، لم يقم فقط بالسَّماح بتقارب بروتوكول OSPF عبر شبكة SDN ولكن أيضاً خلال زمن أفضل مقارنة بالزمن اللازم للتقارب ضمن الشَّبكة التَّقليديَّة. الشَّكل رقم 3 يعطي تمثيل بياني للبيات الواردة بالجَّدول رقم 2:



الشكل 3 مخططّ مُقارنة زمن تقارب بروتوكول OSPF

تم مُحاكاة الشَّبكة الهجينة باستخدام برنامج Mininet وتمَّت إضافة MiniNExT لذلك المُحاكي بهدف محاكاة عقد الشَّبكة التَّقليديَّة للحصول على وقت التَّقارب

ضمن الشَّبكة الهجينة وفق الآليَّة المذكورة سابقاً، أما زمن التَّقارب ضمن الشَّبكة التَّقليديَّة فقد تم تحديده بشكل نظري بالاعتماد على [13]. ساعد ذلك على الحصول على النَّتائج من خلال مثال عملي تم تطبيقه بمساعدة برنامج المحاكاة سابق الذِّكر وذلك بالنسبة لطوبولوجيا الشَّبكة الهجينة إذ تم بناء عدة نماذج من الشَّبكة الهجينة وفي كل نموذج يتم زيادة عدد مبدلات OF كما هو مذكور بالجدول رقم 2 ويُحسب زمن تقارب برتوكول OSPF في كل نموذج شبكة هجينة (بالاعتماد على الوظائف البرمجيَّة للمتحكم والمُبدلات) إذ تم ملاحظة أن ذلك الزَّمن لا يزداد بازدياد عدد المُبدلات كما هو الحال في الشَّبكة التَّقليديَّة والَّتي تم حساب زمن التَّقارب فيها بشكل نظري.

V. الخاتمة

إن الشَّبكات الهجينة المُعرَّفة برمجياً HSDN هي حالة ضروريَّة لانتقال الشَّبكات التَّقليديَّة إلى الشَّبكات المُعرَّفة برمجياً. يحتاج ذلك النَّموذج من الشَّبكات الهجينة لمُتحكِّم هجين قادر على إدارة وظائف الشَّبكة الهجينة. تمَّ في هذا العمل تحديد النَّموذج المُناسب للشَّبكة الهجينة ودراسة آلية عمل المُتحكِّم الهجين ضمن ذلك النَّموذج.

من ثمَّ تمَّ إجراء الاختبارات لتحديد كفاءة المُتحكِّم الهجين. فقد ساهم المُتحكِّم الهجين بإنشاء علاقات تجاور OSPF بين أجهزة التَّوجيه التَّقليديَّة ومخدم التَّوجيه LRS. وهذه نقطة مهمة فهي تُمكّن المُخدم LRS من معرفة واكتشاف الشَّبكة التَّقليديَّة. وبما أنَّ المُتحكِّم الهجين لا يحتاج لبروتوكول STP لتجنب الحلقات ضمن نطاق شبكة SDN، فإنَّ الزَّمن اللازم لإنشاء علاقات تجاور OSPF يكون أقل مما هو عليه ضمن الشَّبكات التَّقليديَّة، ويبقى هذا الزَّمن ثابت مهما زاد عدد مُبدلات OF.

يُمكن أن تُركّز الأعمال المُستقبليَّة على استثمار نموذج الشَّبكة الهجينة المُقترح لتطوير عمل وظائف المراقبة والإدارة ضمن الشَّبكة التَّقليديَّة. أو تحسين عمل برتوكولات أخرى (مثل BGP) أو غيرها. وذلك بالاستفادة من بُنية المُتحكِّم الهجين والذي لديه معلومات مركزيَّة عن شبكة SDN والشَّبكة التَّقليديَّة.

topics in software defined networks - HotSDN '12. New York, USA: ACM Press.

[12] Stringer, J. P., Fu, Q., Lorier, C., Nelson, R., & Rothenberg, C. E. (2013). Cardigan: Deploying a Distributed Routing Fabric**. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN '13. New York, USA: ACM Press.**

[13] Understanding and Tuning Spanning Tree Protocol Timers**. *(2021, July 8). Cisco. Retrieved June 28, 2022, from https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/19120-22.html***

[14] Yang, L., Dantu, R., Anderson, T., and R. Gopal, Forwarding and Control Element Separation (ForCES) Framework**, RFC 3746, , April 2004, <https://www.rfc-editor.org/info/rfc3746>.**

Reference

[1] Caria, M., Das, T., Jukan, A., & Hoffmann, M. (2015). Divide and conquer: Partitioning OSPF networks with SDN**. 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), 467-474.**

[2] Chemalamarri, V. D., Nanda, P., & Navarro, K. F. (2015). Symphony - A controller architecture for Hybrid Software Defined Networks**. 2015 Fourth European Workshop on Software Defined Networks, pp. 55-60.**

[3] *Choi, T., Song, S., Park, H., Yoon, S., & Yang, S. (2014).* SUMA: Software-defined Unified Monitoring Agent for SDN**. *2014 IEEE Network Operations and Management Symposium (NOMS). Krakow, Poland: IEEE.***

[4] Clarke, J., Salgueiro, G., & Pignataro, C. (2016). Interface to the routing system (I2RS) traceability: Framework and Information Model**.**

[5] Fernando, F., João, S., Pedro, V., & Antônio, A. (2013). Integrating legacy forwarding environment to OpenFlow/SDN control plane, **2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS) (15th ed., pp. 1–3). Ieee.**

[6] Installing POX — POX Manual Current documentation. **(2015). Retrieved June 20, 2022, from noxrepo.github.io website: https://noxrepo.github.io/pox-doc/html/**

[7] Lei He, Xiaoning Zhang, Zijing Cheng, & Yajie Jiang. (2016). Design and implementation of SDN/IP hybrid space information network prototype**. 2016 IEEE/CIC International Conference on Communications in China (ICCC Workshops). Chengdu, China: IEEE.**

[8] Levin, D., Canini, M., Schmid, S., & Feldmann, A. (2013). Incremental SDN deployment in enterprise networks. **ACM SIGCOMM Computer Communication Review, 473–474. Hong Kong, China: Association for Computing Machinery (ACM).**

[9] Lu, H., Arora, N., Zhang, H., Lumezanu, C., Rhee, J., & Jiang, G. (2013). HybNET**. Proceedings of the Industrial Track of the 13th ACM/IFIP/USENIX International Middleware Conference on - Middleware Industry '13. New York, USA: ACM Press.**

[10] Quagga Software Routing Suite**. (2017). Retrieved June 20, 2022, from https://www.quagga.net/**

[11] Rothenberg, C. E., Nascimento, M. R., Salvador, M. R., Corrêa, C. N. A., Cunha De Lucena, S., & Raszuk, R. (2012). Revisiting routing control platforms with the eyes and muscles of software-defined networking. **Proceedings of the first workshop on Hot**