

Software Defined Networking (SDN)

Shaban Esmail *

Batoul Ali *

(Received 13 / 6 / 2022. Accepted 13 / 6 / 2022)

□ ABSTRACT □

Software Defined Networking (SDN) is a concept which provides the network operators and data centres to flexibly manage their networking equipment using software running on external servers. According to the SDN framework, the control and management of the networks, which is usually implemented in software, is decoupled from the data plane. On the other hand cloud computing materializes the vision of utility computing. Tenants can benefit from on-demand provisioning of networking, storage and compute resources according to a pay-per-use business model.

Keywords: software-defined networking , control plane , data plan , OpenFlow, Floodlight

طالب ، قسم هندسة الاتصالات والإلكترونيات، كلية الهندسة الميكانيكية والكهربائية، جامعة تشرين، اللاذقية، سورية
Shaban.esmaiel99@gamil.com .

طالب ، قسم هندسة الاتصالات والإلكترونيات، كلية الهندسة الميكانيكية والكهربائية، جامعة تشرين، اللاذقية، سورية*
Shaban.esmaiel99@gamil.com .

(تاريخ الإيداع ٢٠٢٢ / ٧ / ١٣ . قُبل للنشر في ٢٠٢٢ / ٧ / ١٣)

□ ملخص □

الشبكات المعرفة بالبرمجيات (SDN) هي مفهوم يوفر لمشغلي الشبكات ومراكز البيانات إدارة مرنة لمعدات الشبكات الخاصة بهم باستخدام برنامج يعمل على خوادم خارجية. وفقاً لإطار عمل SDN ، يتم فصل التحكم في الشبكات وإدارتها ، والتي يتم تنفيذها عادةً في البرامج ، عن مستوى البيانات. من ناحية أخرى ، تجسد الحوسبة السحابية رؤية حوسبة المرافق. يمكن للمستأجرين الاستفادة من توفير الشبكات والتخزين وحساب الموارد عند الطلب وفقاً لنموذج أعمال الدفع لكل استخدام.

الكلمات المفتاحية: الشبكات المعرفة بالبرمجيات ، طبقة التحكم ، طبقة البيانات ، بروتوكول

مقدمة: الشبكة مجموعة من الأجهزة المتصلة مع بعضها البعض عن طرُق أجهزة شبكة مثل : مبدلات وموجهات (Router -switch) ، وعلم الشبكات من أهم العلوم التي تهتم بتبادل المعلومات ونقلها ومشاركتها من مكان إلى آخر بشكل آمن و إلى الوجهة المناسبة . ففي الشبكات التقليدية كانت تتم هذه العملية من خلال المكونات الأساسية للشبكة من switch و Router ، حيث كان كل جهاز شبكي منها يعتمد على ثلاث عناصر :

plane data –plane Control و البيانات نفسها ، وهذا هو السبب الذي جعل هذه الشبكات معقدة و غالية ،حيث كان يقع على عاتق هذين المكونين كل عمليّة التوجيه و إقرار التوجيه اختيار المسارات المناسبة بأقل التكاليف و بأقصر زمن ممكن ، مع استمرار العمل بهذا النمط فترة من الزمن دون أن يطرأ أي تعديل على البنية التحتية رغم أهمية الشبكة في التقدم في مجال تكنولوجيا المعلومات إلى أن جاءت تقنية ال Defined Software: SDN networking في محاولة ناجحة لإنهاء هذا العجز ،وإحداث تغيير جذري في البنية التحتية للشبكات بما يتناسب مع التطور التقني ، حيث اعتمدت هذه التقنية على تغيير معمارية الشبكات ، وتم في هذه التقنية فصل مستوى التحكم عن أجهزة الشبكة ،وجعل عمل هذه الأجهزة مقتصرًا على تمرير البيانات فقط ، وعملت هذه التقنية على حل المشاكل المتعلقة بإدارة الشبكة ، وقللت من التعقيد البرمجي الذي كان في الشبكة التقليدية . لاحظ الجميع منذ بضع سنوات التحسن المستمر في سرعة الإنترنت حيث بلغ معدل نقل البيانات من ١٠ إلى ١٠٠ جيجا بايت في الثانية الواحدة ، ومع وجود خوادم عملاقة وظهور الحوسبة السحابية فقد تم حل مشكلة تخزين و معالجة البيانات ، أما عناوين الإنترنت فقد انتهت مع ظهور عناوين الإصدار السادس IPv6 ، ولكن بقيت إحدى المشاكل التي يريد الجميع حلها وهي الاعتماد على بنية تحتية قوية مؤمنة بسيطة ذات تكلفة منخفضة وسهل التعامل معها ، وحل مشاكل الإدارة في الشبكة والتعقيد والكلفة المرتفعة للعتاد الحقيقي في حال كان يقدم ما يريده المستخدم ومن هنا بدأ البحث عن حلول بديلة وظهرت تقنية ال SDN فهي باختصار الجيل القادم للبنية التحتية في هندسة الشبكات لأنها تقوم بعمل لا نستطع عمله مع العتاد التقليدي الذي نستخدمه حاليا .

أهمية البحث وأهدافه:

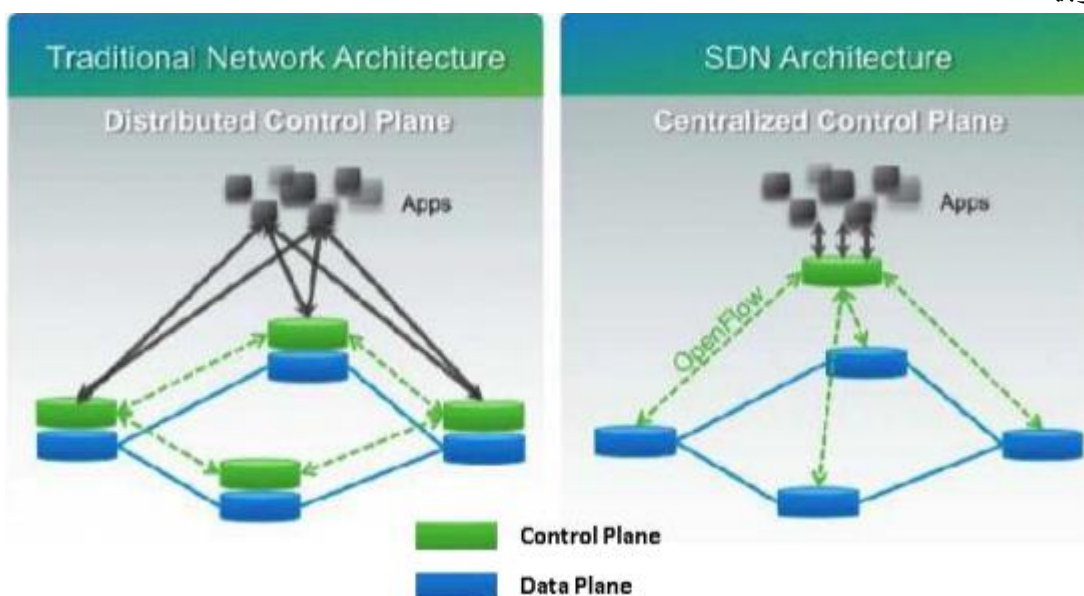
طبقت سيناريوهات المحاكاة باستخدام متحكم floodlight مدمج مع محاكي mininet باستخدام نظام Ubuntu وتمت دراسة الانتاجية باستخدام الأداة ipref و كذلك تمت دراسة التأخير باستخدام الأداة Cbench

مقدمة :

تعرف تقنية ال SDN: اختصار ل Networking Defined software هذه التقنية التي من خلالها تم فصل plane Data عن ال plan control في أجهزة الشبكة لتصبح دور هذه الأجهزة مقتصر على تمرير البيانات أما الإدارة والتحكم ستصبح في طبقات جديدة

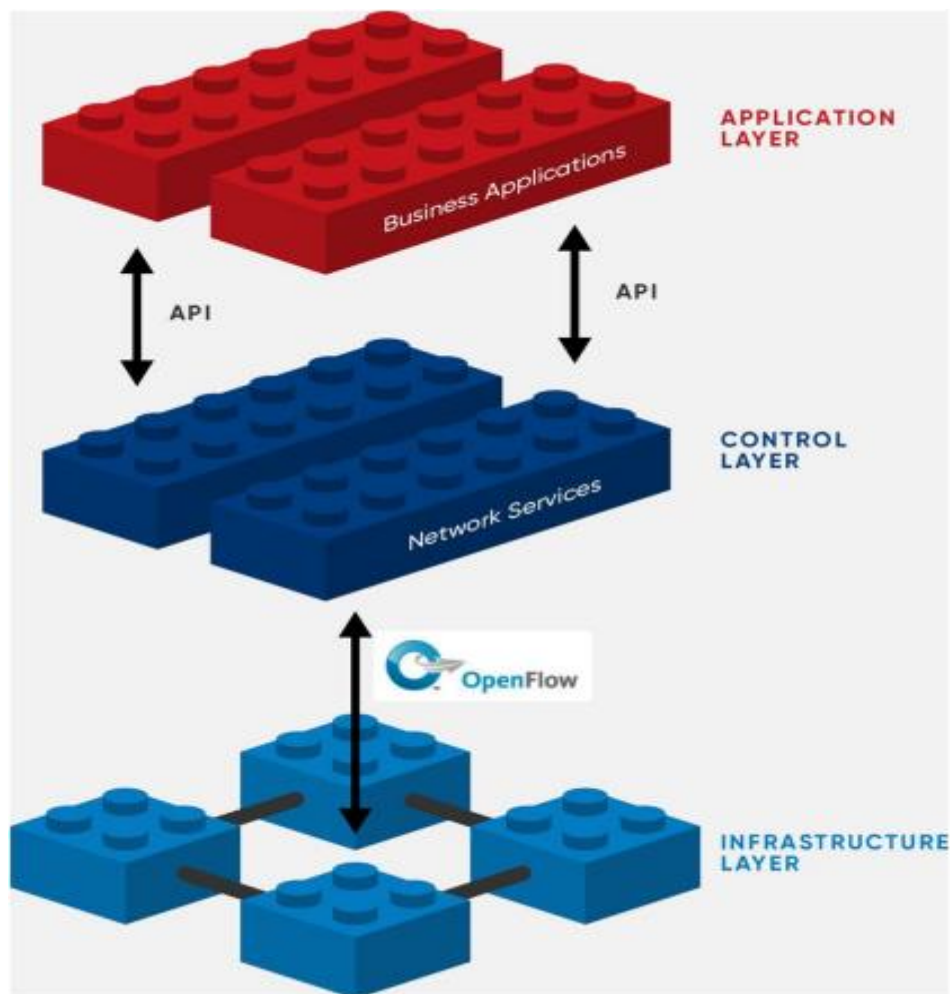


في الشبكات التقليدية تكون عملية تمرير البيانات وتقرير وجهة البيانات (التوجيه) متضمنان ضمن نفس الجهاز وهذا ما جعلها معقدة . في تقنية ال SDN فإن كل المتطلبات الخاصة بتوجيه البيانات تم إنجازها في برمجية خاصة (المتحكم) ، أما الأجهزة فهي ببساطة تطبق القرارات التي يرسلها لها المتحكم على رزم البيانات ، ولهذا السبب فإنه لا حاجة لأن يكون لأجهزة الشبكة إدراك لمنطق التوجيه ، إنما فقط تخزين caching لقرارات المتحكم بشأن التوجيه .



يتكون مستوى التحكم من وحدة تحكم أو أكثر ، والتي تعتبر دماغ شبكة ال SDN حيث يتم فيها دمج الذكاء بالكامل ، ومع ذلك فإن المركزية الذكية لها عيوبها الخاصة عندما يتعلق الأمر بالأمن وقابلية التوسع والمرونة وهذه هي القضية الرئيسية في ال SDN .

البنية التحتية للشبكات المعرفة بالبرمجيات ال SDN :
تقسم الى ثلاث طبقات موضحة في الشكل التالي :



1 (طبقة التطبيقات Layer Application :

تتكون من الخدمات والتطبيقات التي تقدمها الشبكة للمستخدم (كالتوجيه وجودة الخدمة)، وتتواصل مع الطبقة التي تليها (طبقة التحكم) عن طريق واجهات برمجية (API(Application Programming Interface) وهي واجهات يتم استخدامها من قبل المهندسين لكي تساعد الشبكة على تقديم خدماتها وتطبيقاتها ببرمجتها ، لذلك يقال أن ال SDN مبنية على أساس البرمجة.

2 (طبقة التحكم Control Layer :

تمثل نقطة التحكم المركزية point central control بأجهزة الشبكة حيث يتم التحكم في الشبكة وإدارتها عن طريق جهاز أساسي يسمى ال Controller ، حيث يتم التواصل بين المتحكم وأجهزة الشبكة عن طريق بروتوكول خاص بال SDN هو flow open : OF :

3 (طبقة البنية التحتية /التوجيه Infrastructure/ forwarding Layer :

تقوم بتمرير البيانات إلى وجهاتها.

متحكمات الشبكات المعرفة بالبرمجيات SDN Controller :
يعتبر المتحكم هو أساس شبكات ال SDN وهو سبب تطوير هذه التقنية ، حيث يعتبر نقطة التحكم المركزية والمرجعية في هذه التقنية ، ويمثل عقل الشبكة يقوم بإصدار القواعد التي تنفذها الأجهزة الموجودة في طبقة ال Data plane يتواصل مع هذه الطبقة عبر الجسر الشمالي .
بعض أنواع المتحكمات:



1 (NOX :

مبني بلغة ++C بالاعتماد على Boost

2 (Rue controller :

متحكم مفتوح المصدر مكتوب بلغة البايثون python يدعم عدة إصدارات من بروتوكول ال OF ويدعم بروتوكول ال STP وبالتالي يمنع حدوث حلقات

3 (Beacon :

هو متحكم مفتوح المصدر مبنية بلغة الجافا Java ويدعم بروتوكول ال OF

4 (ONOS: Open Network Operating System :

هو النظام الأساسي الوحيد الذي يدعم الانتقال من الشبكات القديمة إلى شبكات SDN وهو متحكم مفتوح المصدر مكتوب بلغة ال Java

5 (POX :

يعتبر من المتحكمات الشهيرة والهامة في شبكات ال SDN ، وقد تمت برمجته بلغة البايثون Python ، كما أنه يوفر منصة برمجية سهلة ومرنة لكتابة التطبيقات التي يمكن تنفيذها في الشبكة . تتناول أغلب الأبحاث في مجال ال SDN المتحكم POX لتنفيذ عدة تطبيقات مثل موازنة الحمل وتطبيقات الوسائط المتعددة ونقل الملفات ومخدم ويب وغير ذلك من التطبيقات ، مما يتيح إمكانية التطوير المستمر لأداء الشبكة.

6 (Flood Light :

متحكم مفتوح المصدر يستعمل في شبكات ال SDN ذو واجهة ويب WEB مبنية بلغة جافا وأحد أكثر أنواع المتحكمات شيوعا .

بروتوكول ال OF(Open Flow) :

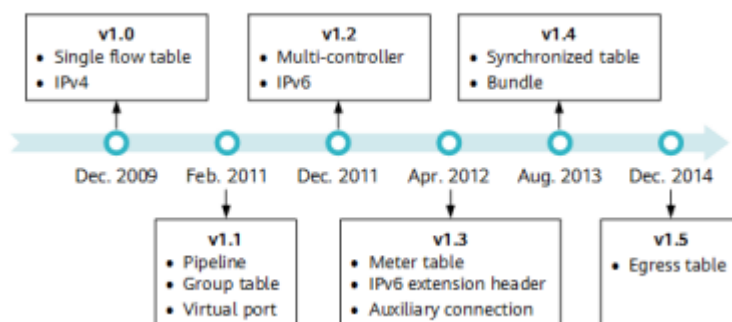
هو بروتوكول التواصل بين طبقة التحكم control layer وطبقة البنية التحتية infrastruction layer التي تليها . تم اعتماد النسخة الأولى منه عام 2011 (open flow v1.1) ثم تابعت منظمة ال OFN تطويره حتى تم إطلاق النسخة الأخيرة منه عام 2014 (open flow v1.105)

يعتبر هذا البروتوكول التطبيق العملي لفكرة شبكات ال SDN ، ويرى بعض الباحثين شبكات ال SDN وبروتوكول OF مفهوم واحد ، وهو من عائلة ال Tcp ويستخدم المنفذ 6653

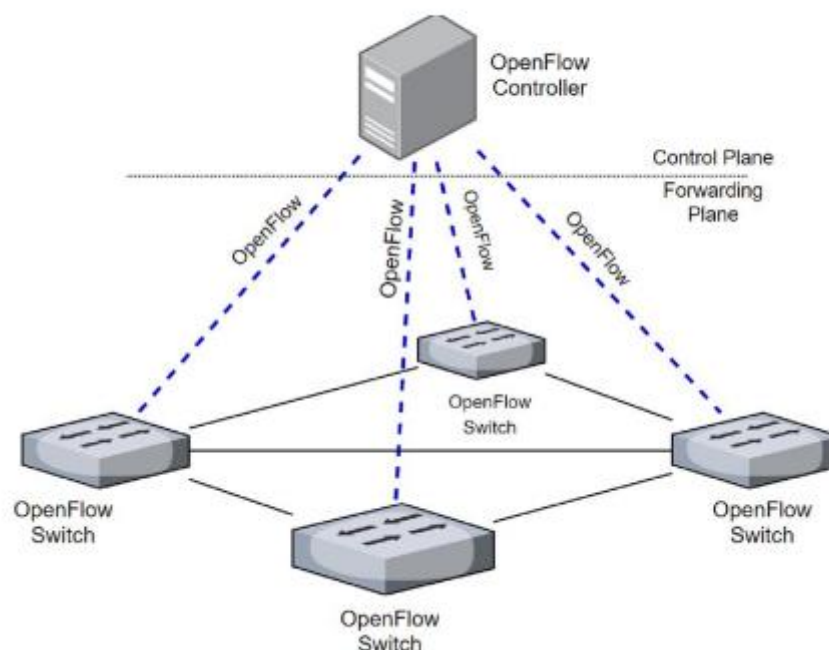
يعمل هذا البروتوكول في الطبقة الثانية من معمارية شبكات ال SDN أما حسب توزيع الطبقات في الشبكات التقليدية فيمكن القول أنه يعمل في الطبقة الثالثة ؛ وفي الحالتين تكمن وظيفته في نقل أوامر التوجيه وتحديد المسارات من طبقة التحكم (CP)

control plane إلى طبقة البنية التحتية Data plan (DP) ، و أيضا نقل متطلبات البنية التحتية إلى مركز القيادة

وبالتالي عند إنشاء طوبولوجيا أو إضافة مبدل إلى الشبكة فإن المبدل هو الذي يقوم بتأسيس الاتصال مع المتحكم عن طريق بروتوكول التدفق المفتوح ، وبما أن هذا البروتوكول يعمل بنمط ال TCP بالتالي سوف تجري عملية ال hand checking والتي هي عبارة عن رسائل من المبدل إلى المتحكم ، ويتم الرد عليها من قبل المتحكم ومن خلال هذه الرسائل يتم التعرف على نسخة ال OF التي سوف نتعامل معها وذلك بسبب تعدد إصدارات هذا البروتوكول .



ويقوم ال OF بتخزين قواعد التوجيه لدى أجهزة الشبكة في جداول التدفق Flow Entries ؛ ويعرف التدفق بأنه الرزم packets الواردة إلى منفذ معين ، ويعرف مدخل التدفق Flow Entry بأنه الحدث أو التعليمات الواجب على هذه الرزم الواردة وتشمل إما إضافة أو تعديل أو حذف مسار موجود وذلك وفقا لما يقرره المتحكم . يوضح الشكل التالي آلية عمل جهاز من الشبكة إذ يتلقى الأوامر والتوجيهات من طبقة التحكم و يقوم بنقل المتطلبات إليها عن طريق واجهة برمجية لبروتوكول Open Flow.



عناصر البنية التحتية في ال Open Flow

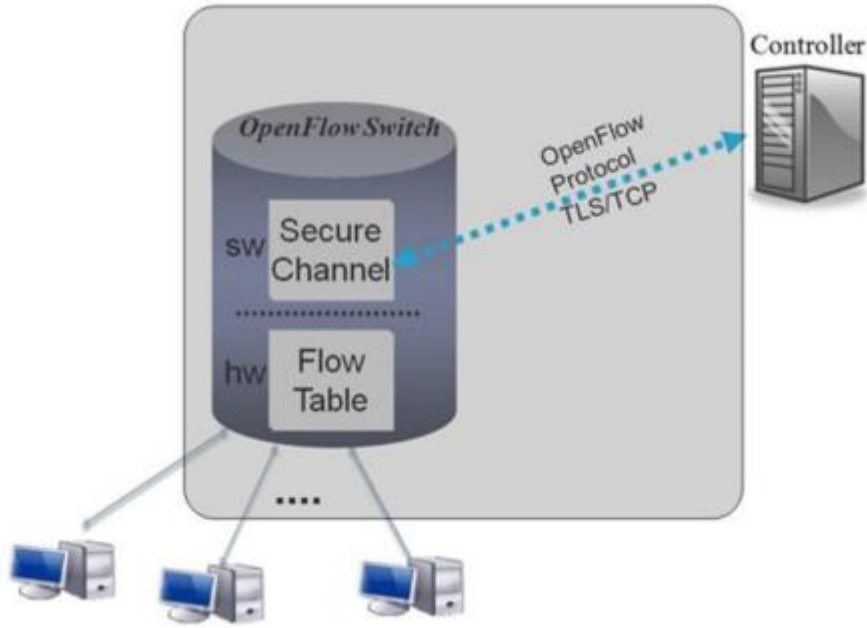
تتألف البنية التحتية للشبكات التي تستخدم بروتوكول OF من عدد من التجهيزات أو المبدلات التي تعمل وفق قيادة متحكم مركزي

1 Open Flow Switches

يتألف هذا المبدل من واحد أو أكثر من جداول التدفق ، حيث تقوم المبدلات بإنجاز عمليات البحث والتوجيه ، ويتم إدارة هذه الجداول من قبل المتحكم الذي يتصل بقناة آمنة مع ال switch ، يتألف كل جدول من عدد من المداخل الذي يعرف القواعد التي تتألف كل قاعدة منها من ال header العداد ومجموعة الأفعال المترافقة مع كل فعل جديد .

2 Open Flow channel (secure channel)

هي واجهة تربط مكونات open flow مع المتحكم controller ، فمن خلال هذه الواجهة يقوم المتحكم بإدارة وبرمجة جداول التدفق Flow tables الموجودة في المبدل ، ويتم تشفير لقناة بواسطة طبقة النقل TLS: Transport Layer Security ، لكن يمكن العمل مباشرة بواسطة بروتوكول التحكم بالنقل TCP .



الفوائد التي قدمتها الشبكات المعرفة بالبرمجيات SDN :

- 1 (قابلية البرمجة بشكل مباشر ، والتحكم بشكل مباشر ايضا .
- 2 (يستطيع مهندس الشبكة التحكم بالشبكة بشكل كامل من مكان واحد حيث يقوم بعمل إدارة وصيانة من مكان واحد .
- 3 (تحسين عملية إرسال البيانات في الشبكة من ناحية التوجيه و توزيع الحركة في الشبكة .
- 4 (سهولة صيانة الشبكة ومراقبة الشبكة بشكل أوسع وأسهل ، حيث أنه يتواجد وحدة مركزية للتحكم الكامل في الشبكة كلها
- 5 (توفير عدد كبير من أجهزة الشبكة ، حيث أنه نستطيع عمل أجهزة افتراضية ولكن وهمية غير موجودة في الواقع
- 6 (ستتواجد شبكات افتراضية برمجية وستكون أسهل بكثير من أن يكون عدة شبكات موجودة في الواقع الحقيقي ، حيث أنه سيوفر لنا الكثير من الوقت وتوفير من ناحية التكلفة وسهولة الإدارة .
- 7 (يستطيع مهندس الشبكة توسيع الشبكة بكل سهولة بشكل افتراضي وهذا يسهل الكثير من العمل على مهندسي الشبكة ويكون أفضل من أن تكون الشبكة موجودة بشكل حقيقي .

الميزات التي تقدمها الشبكات المعرفة بالبرمجيات من الجانب الأمني

تم مؤخراً إجراء الكثير من الأبحاث لتوفير الأمان الجوهري لشبكة بواسطة شبكة ال SDN لذا قاموا بتعميم خصائص شبكة ال SDN ، وكذلك الشبكات التقليدية ، والتي تؤثر على الأمن الذي توفره الشبكة من أجل مقارنة أفضل . تم تلخيص نتائج التعميم في الجدول

1 (إدارة الشبكة : على النقيض من الشبكات التقليدية ، توفر ال SDN رؤية مركزية ، وبالتالي فإن إدارة SDN أبسط من إدارة الشبكات التقليدية، الصيانة أكثر مرونة مما يوفر التكاليف والوقت لمعالجة الأخطاء ويبسط نشر سياسات الشبكة ككل.

يعد دمج تطبيقات الأمان الجديدة (مثل جدار الحماية) أسهل في شبكة ال SDN نظراً للمرونة والرؤية العالمية في دمج التطبيقات القديمة ، من السهل أيضاً دمج التطبيقات القديمة ، لكن بالإضافة إلى ذلك فإن ال SDN توفر من خلال نظرتها العالمية على الشبكة ومرونتها فيما يتعلق بالصيانة وإعادة التكوين بيئة أمان طبيعية لمواجهة التحديات تم تقييم إدارة شبكات ال SDN على أنها إيجابية، لأنها أكثر مرونة من الشبكات التقليدية بسبب النظرة العامة ، وتم تقييم إدارة الشبكات التقليدية على أنها سلبية

2 (إدارة الشبكة :

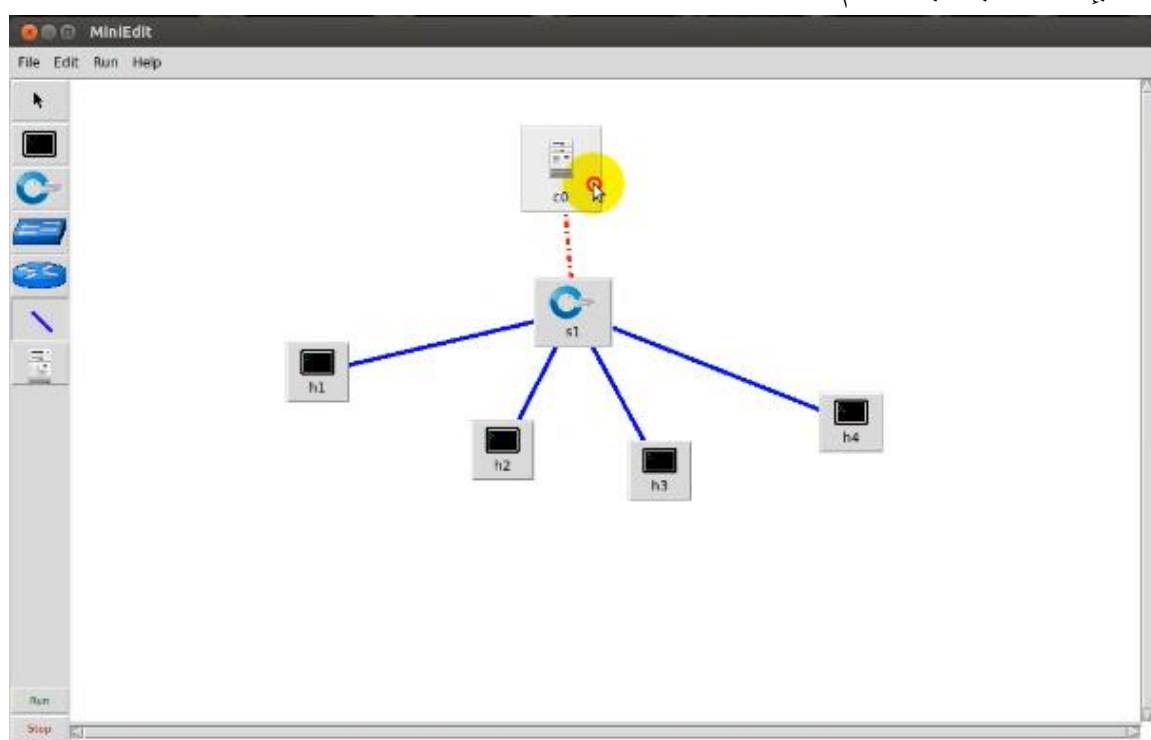
إن الصيانة السهلة لشبكات ال SDN لها تأثير جانبي في توفير التكاليف ، من ناحية تكاليف الموظفين لأن معالجة الأخطاء أسهل وأقل جهد، إضافة إلى ذلك يمكن أن يؤدي نشر شبكات ال SDN إلى تقليل استهلاك الطاقة للأجهزة من ناحية ، ولم تعد السويتشات مسؤولة عن مهام حسابية قوية من ناحية أخرى . لذا يتم تقييم هذا المعيار ل SDN على أنه إيجابي حيث يتم تخفيض الأجهزة ويتناقص استهلاك الطاقة كما يتم توفير تكاليف الصيانة ، على العكس من ذلك تم التقييم في الشبكات التقليدية على أنه سلبي بسبب تعقيد الدارة . العيب المهم في طوبولوجيا شبكة ال SDN هو نقطة الفشل الوحيدة في حال عدم توفر وحدة التحكم ؛ولهذا السبب تم تقييم المتانة في الشبكات المعرفة بالبرمجيات بأنها سلبية، لذا تعد الشبكات التقليدية أكثر قوة ضد الانقطاع بسبب لا مركزيتها ، وهذا هو السبب في أننا نقيم قوة الشبكات التقليدية على أنها إيجابية

3 (كشف الهجوم وتخفيفه :

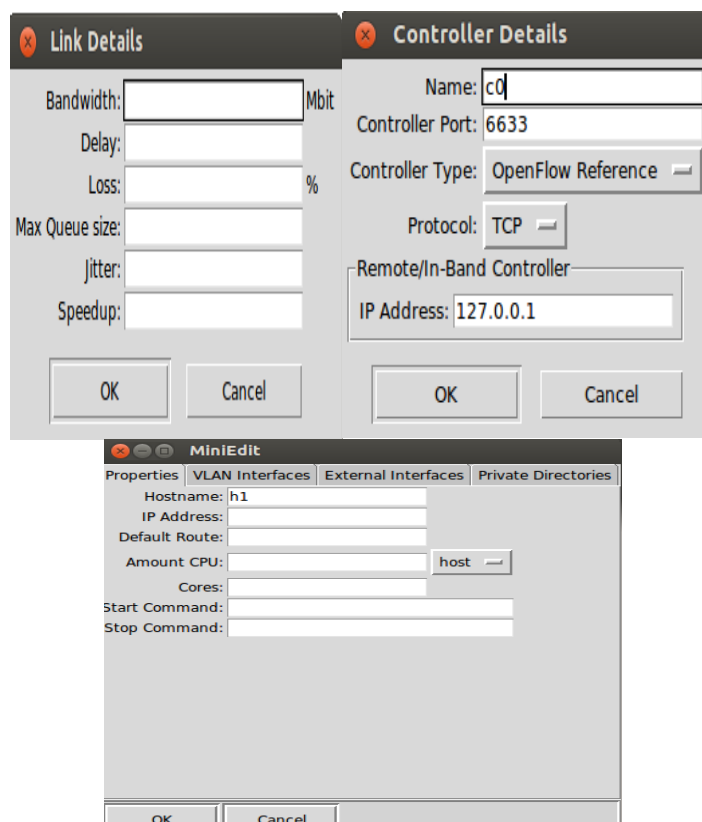
لا تزال ال SDN قيد التطوير ولا يتم نشرها على نطاق واسع ، مما يسمح لمشغلي الشبكات بدمج عملية الكشف عن هجمات الشبكة والتخفيف منها حسب التصميم، نتيجة هذا تم تقييم إمكانية اكتشاف /تخفيف الهجوم في ال SDN على أنه إيجابي ، وتم تقييم هذا المعيار للشبكات التقليدية على أنه سلبي لأن تطبيق التغيرات على هذه الشبكات التقليدية أكثر تعقيداً.

النتائج والمناقشة:

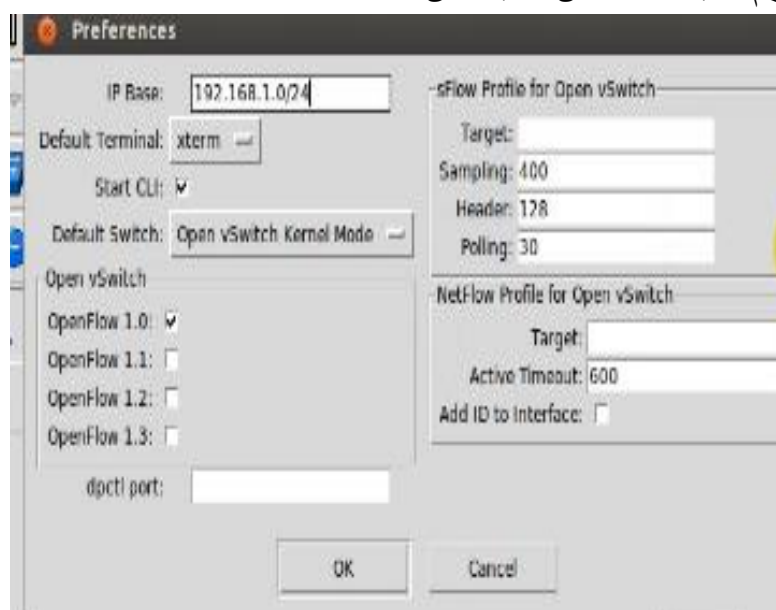
قمنا بإنشاء الشبكة باستخدام الاداة mini edit :



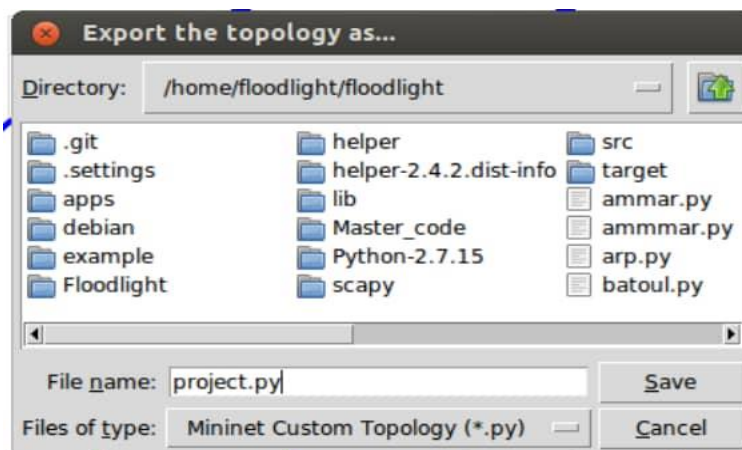
وتم ضبط خصائص الوصلات و المتحكم و المضيف



وتم ضبط خصائص الشبكة من خلال



وتم حفظ الشبكة :



- نقوم بتشغيل المتحكم floodlight
- ندخل إلى ملف التنصيب باستخدام الأمر :
- cd floodlight
- ولتشغيل المتحكم نستخدم الأمر :
- java -jar target/floodlight.jar

```
Floodlight@floodlight:~$ cd floodlight/
Floodlight@floodlight:~/floodlight$ java -jar target/floodlight.jar
14:22:08.305 INFO [n.f.c.m.FloodlightModuleLoader:main] Loading modules from src/main/resources/floodlightdefault.properties
14:22:08.403 ERROR [n.f.c.m.FloodlightModuleLoader:main] Could not find module: net.floodlightcontroller.core.module.IFloodlightM
odule: Provider net.floodlightcontroller.mactracker.MACTracker not found
14:22:08.406 WARN [n.f.r.RestApiServer:main] HTTPS disabled; HTTPS will not be used to connect to the REST API.
14:22:08.406 WARN [n.f.r.RestApiServer:main] HTTP enabled; Allowing unsecure access to REST API on port 8080.
14:22:09.138 WARN [n.f.c.i.OFSwitchManager:main] SSL disabled. Using unsecure connections between Floodlight and switches.
14:22:09.138 INFO [n.f.c.i.OFSwitchManager:main] Clear switch flow tables on initial handshake as master: TRUE
14:22:09.138 INFO [n.f.c.i.OFSwitchManager:main] Clear switch flow tables on each transition to master: TRUE
14:22:09.144 INFO [n.f.c.i.OFSwitchManager:main] Setting 0x4 as the default max table to receive table-miss flow
14:22:09.144 INFO [n.f.c.i.OFSwitchManager:main] Setting max table to receive table-miss flow to 0x4 for DPID 00:00:00:00:00:0
0:01
14:22:09.145 INFO [n.f.c.i.OFSwitchManager:main] Setting max table to receive table-miss flow to 0x4 for DPID 00:00:00:00:00:0
0:02
14:22:09.146 INFO [n.f.c.i.Controller:main] Controller role set to ACTIVE
14:22:09.162 INFO [n.f.f.Forwarding:main] Default hard timeout not configured. Using 0.
14:22:09.162 INFO [n.f.f.Forwarding:main] Default idle timeout not configured. Using 5.
14:22:09.162 INFO [n.f.f.Forwarding:main] Default priority not configured. Using 1.
14:22:09.162 INFO [n.f.f.Forwarding:main] Default flags will be empty.
14:22:09.162 INFO [n.f.f.Forwarding:main] Default flow matches set to: VLAN=true, MAC=true, IP=true, TPPT=true
14:22:09.322 INFO [o.s.s.i.c.FallbackCCPProvider:main] Cluster not yet configured; using fallback local configuration
14:22:09.323 INFO [o.s.s.i.SyncManager:main] [32767] Updating sync configuration ClusterConfig [allNodes={32767=Node [hostname=lo
calhost, port=6642, nodeId=32767, domainId=32767]}, authScheme=CHALLENGE_RESPONSE, keyStorePath=/etc/floodlight/auth_credentials.
jceks, keyStorePassword is unset]
14:22:09.389 INFO [o.s.s.i.r.RPCService:main] Listening for internal floodlight RPC on localhost/127.0.0.1:6642
14:22:09.397 INFO [n.f.c.i.OFSwitchManager:main] Listening for switch connections on 0.0.0.0/0.0.0.0:6653
14:22:09.401 INFO [n.f.l.i.LinkDiscoveryManager:main] Setting autoportfast feature to OFF
14:22:12.860 INFO [n.f.j.JythonServer:debugserver-main] Starting DebugServer on :6655
14:22:24.468 INFO [n.f.l.i.LinkDiscoveryManager:Scheduled-1] Sending LLDP packets out of all the enabled ports
```

- تستخدم الأداة Cbench من أجل دراسة بارامترات التأخير والانتاج لمتحكم SDN
- تحاكي وجود العديد من المبدلات التي تتصل بالمتحكم وإرسال رسائل له.
- تعمل وفق الخوارزمية التالية:

cbench is a benchmarking tool for controllers

Algorithm:

```
pretend to be n switches (n=16 is default)
create n openflow sessions to the controller
if latency mode (default):
  for each session:
    1) send up a packet in
    2) wait for a matching flow mod to come back
    3) repeat
    4) count how many times #1-3 happen per sec
else in throughput mode (i.e., with '-t'):
  for each session:
    while buffer not full:
      queue packet_in's
      count flow_mod's as they come back
```

- في حال كنا نريد استخدام أداة Cbench لدراسة التأخير في المتحكم Floodlight من أجل دورتين و 16 مبدل و 10 مضيفين لكل مبدل والتتفيذ دورتين حيث نلاحظ أن متوسط التأخير هو 7356.99 responses/s

```
floodlight@floodlight:~/floodlight$ cbench -c localhost -s 16 -M 10 -l 2 -p 6653

cbench: controller benchmarking tool
  running in mode 'latency'
  connecting to controller at localhost:6653
  faking 16 switches offset 1 :: 2 tests each; 1000 ms per test
  with 10 unique source MACs per switch
  learning destination mac addresses before the test
  starting test with 0 ms delay after features_reply
  ignoring first 1 "warmup" and last 0 "cooldown" loops
  connection delay of 0ms per 1 switch(es)
  debugging info is off
01:52:04.366 16 switches: flows/sec: 105 436 127 355 147 83 200 77 228
21 271 167 310 133 137 56 total = 2.852940 per ms
01:52:05.467 16 switches: flows/sec: 460 521 497 470 457 500 501 425 4
48 485 476 421 432 471 434 359 total = 7.356985 per ms
RESULT: 16 switches 1 tests min/max/avg/stdev = 7356.99/7356.99/7356.99/0.00 res
ponses/s
```

- ومن أجل حساب الانتاجية :
- حيث نلاحظ أن متوسط الانتاجية هو
- 71.491responses/s

```
floodlight@floodlight:~/floodlight$ cbench -c localhost -s 16 -M 10 -l 2 -p 6653 -t
cbench: controller benchmarking tool
  running in mode 'throughput'
  connecting to controller at localhost:6653
  faking 16 switches offset 1 :: 2 tests each; 1000 ms per test
  with 10 unique source MACs per switch
  learning destination mac addresses before the test
  starting test with 0 ms delay after features_reply
  ignoring first 1 "warmup" and last 0 "cooldown" loops
  connection delay of 0ms per 1 switch(es)
  debugging info is off
01:54:40.737 16 switches: flows/sec: 0 0 0 6418 0 0 0 0 804 3212 0 0 711
1 0 4805 0 total = 22.349754 per ms
01:54:41.839 16 switches: flows/sec: 3594 0 0 18847 0 0 0 0 10392 11137 0
0 10123 0 17399 0 total = 71.491929 per ms
RESULT: 16 switches 1 tests min/max/avg/stddev = 71491.93/71491.93/71491.93/0.00 responses/s
```

- يمكن قياس التأخير في شبكة SDN وبرنامج mininet من خلال رسائل ping
- مثلاً لنأخذ الشبكة

Sudo mn

ومن ثم سوف نقوم باستخدام الأمر

h1 ping -c3 h2

```
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.89 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.381 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.029 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.029/1.435/3.897/1.746 ms
mininet>
```

دراسة الإنتاجية :

- نقوم بإنشاء الشبكة باستخدام الأمر :

Sudo mn — topo single,2

- نقوم بالدخول إلى المستخدمين h1 و h2 من خلال الأمر

Xterm h1 h2

- نقوم بتشغيل أحد المضيفين وليكن h2 كمخدم من خلال الأمر
- نقوم بتشغيل المضيف الآخر h1 كزبون من خلال الأمر
- `lperf -s -p 5566 -i1 > result`

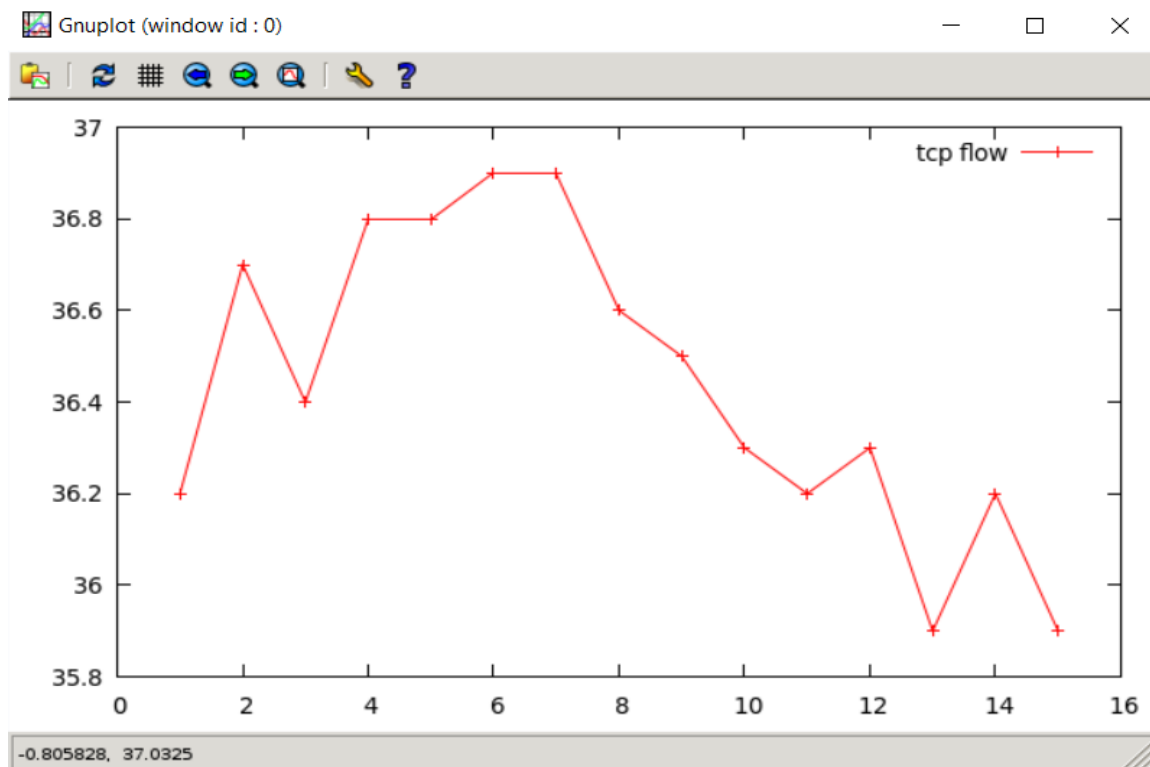
- نقوم بتشغيل المضيف الآخر h1 كزبون من خلال الأمر
- `iperf-c 10.0.0.2 -p 5566 -t 15`
- عملية المعالجة من خلال الأمر

`cat result | grep sec | head -15 | tr - " " | awk '{print $4,$8}' > new_result`

- يمكن عرض القيمة من خلال الأمر `more new_result`

```
root@mininet-vm:~# more new_result
1.0 36.9
2.0 37.1
3.0 37.7
4.0 37.8
5.0 37.5
6.0 37.1
7.0 37.4
8.0 37.9
9.0 37.2
10.0 36.2
11.0 33.5
12.0 31.8
13.0 33.7
14.0 37.4
15.0 36.5
root@mininet-vm:~#
```


- من أجل الرسم ندخل إلى gnuplot ومن ثم نستخدم الأمر
- plot "new_result" title "tcpflow" with linespoints



References

- [1] D. Rana and S. Chamoli, "Software Defined Networking (SDN) Challenges, issues and Solution", in International Journal of Computer Sciences and Engineering, February 2019
- [2] floodlight.atlassian.net.available at:
<https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/40403023/w eb+gui>. Last visit 1 August 2019

